2007-09

Fusion of multiple sensor types in computer vision systems

Mayo, Donald R.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/3268

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**FUSION OF MULTIPLE SENSOR TYPES
IN COMPUTER VISION SYSTEMS**

by

Donald Ray Mayo Jr.

September 2007

Thesis Co-Advisors:	Mathias Kolsch Kevin Squire

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Fusion of Multiple Sensor Types in Computer Vision Systems

5. FUNDING NUMBERS

6. AUTHOR(S) Donald R. Mayo Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

This research provides analysis of several approaches to the fusion of multiple dissimilar sensors to supplement simple color vision detection and recognition. Non-visible sensor systems can enhance computer vision systems. Our research investigates using thermal infrared (IR) sensors in combination with color data for object detection and recognition. We analyze several types of high-level and low-level sensor fusion to compare error rates with raw color and raw IR error rates in detection and recognition of vehicles in a scene. Principal components analysis is used to reduce the dimensionality of sensor input data in order to discard non-essential data, while preserving data important to classification. One recognition method showing promise is to exploit the strength of non-visible information (low light, shadows, etc.) to reduce the search space for color data by replacing the V channel in the HSV color sensor data with IR. For detection, one method showing promise is replacement or averaging of the dominant color channel with IR.

14. SUBJECT TERMS Sensor Fusion, Principal Components Analysis (PCA), Infrared Imagery, Computer Vision, Dissimilar Sensor Fusion, Object Detection, Object Recognition, High-Level Fusion, Low-Level Fusion, Vehicle Recognition

15. NUMBER OF PAGES
107

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

FUSION OF MULTIPLE SENSOR TYPES IN COMPUTER VISION SYSTEMS

Donald R. Mayo Jr.
Captain, United States Marine Corps
B.S., Oregon State University, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2007

Author: Donald R. Mayo Jr.

Approved by: Mathias Kolsch
Thesis Co-Advisor

Kevin Squire
Thesis Co-Advisor

Peter Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This research provides analysis of several approaches to the fusion of multiple dissimilar sensors to supplement simple color vision detection and recognition. Non-visible sensor systems can enhance computer vision systems. Our research investigates using thermal infrared (IR) sensors in combination with color data for object detection and recognition. We analyze several types of high-level and low-level sensor fusion to compare error rates with raw color and raw IR error rates in detection and recognition of vehicles in a scene. Principal components analysis is used to reduce the dimensionality of sensor input data in order to discard non-essential data, while preserving data important to classification. One recognition method showing promise is to exploit the strength of non-visible information (low light, shadows, etc.) to reduce the search space for color data by replacing the V channel in the HSV color sensor data with IR. For detection, one method showing promise is replacement or averaging of the dominant color channel with IR.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS

EV	Eigenvector, Eigenvalue (depends on circumstance of use)
GA	Genetic Algorithm
IR	Infrared
LIDAR	Light Detection and Ranging
PCA	Principal Components Analysis
RADAR	Radio Detection and Ranging
ROI	Region of Interest
SIFT	Scale Invariant Feature Transform

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

It is 2304 on a Saturday night in February as I type this dedication. I type the dedication this early, because I know who will be the people who will contribute the most to my success in this endeavor. I know the people who will contribute the most to my success because they are the people who have contributed the most to my success throughout my teenage and adult life.

First and foremost, my thanks and devotion must be to my Lord and Savior Jesus Christ. I started life in a family along side near family members who eventually ended up in prison and dead. I was steered away from that path because of the love of Jesus. This love was often manifested through my grandfather Jerome Bowen. My Grandpa is the toughest and my caring man I have ever known. Without my Grandpa taking care of me the way he did, who knows...

Even with the influence of my Grandpa, I would have been nothing and would have done nothing were it not for my wife, the love of my life. She has put up with so much from me and the Corps that it is amazing that she is still around. Her love makes me a better man, a better father, a better Christian and a better Marine. She is the most caring person I know. She has a tough exterior, to be sure. Still, she would give you her life's blood if it would help. She has poked me, pulled me and prodded me to become more than I could have ever been without her. Truly, most of the time just looking at her and feeling her love made me want to do more, be more. I love you and look forward to many years together. I can't wait to grow old with you.

Any time in grad school is fraught with perils in terms of time management. No one knows this better than my daughter does. Ashlynn has spent far too much time waiting for her daddy to get done with the latest project, paper, experiment, etc... I can only hope that she sees the emphasis that mommy and daddy have put on education and that this propels her to higher heights than even mommy and daddy have been able to attain. Sweetheart, I love you and I have been so privileged to have been your dad. I can't

wait to see the wonderful young lady you grow up to be. Stay strong in the faith, let Jesus be your guide and you will never go wrong.

When writing a thesis, no acknowledgement can be complete without an acknowledgement of the hard work and mentoring of the thesis advisor(s), in my case Mathias Kolsch and Kevin Squire. I came into this thesis with a vague idea of what I wanted to do and what I wanted to learn. Through the considerable efforts of Kevin and Mathias, I have accomplished more than I could have expected or anticipated. Thanks for your patience, understanding and mentoring of this hardheaded Marine.

I. INTRODUCTION

A. AREA OF RESEARCH

To supplement simple color vision detection and recognition, this research aims to provide analysis of several approaches to the fusion of multiple dissimilar sensors. In this thesis, we use an uncooled 8-12 micrometer bolometer thermal IR camera and a color CCD camera.

Our research focused on three main questions:

1. How, and by how much, can object detection and recognition be improved by the addition of out of the visible spectrum sensors to computer vision systems?

2. What are the advantages in terms of object recognition and detection we can gain from out of-the-visible spectrum sensors such as thermal IR, RADAR, LIDAR, chemical sensors, etc…?

3. What is the effectiveness of early versus late binding (low-level versus high-level fusion) of sensor data?

We will attempt to answer these questions by experimentation in detection and recognition of vehicles using different types and combinations of sensor fusion with thermal IR and color sensor data. We hypothesize that using multiple sensors, in a way that best utilizes each sensor's strength, will contribute to better and more rapid recognition and detection of vehicles in cluttered, varying backgrounds.

B. BACKGROUND

Most current computer vision systems center around the use of simple color or grayscale imagery. Many of these techniques, depending on input data type, apply in a relatively straightforward manner to data from sensors that sense parts of the electromagnetic spectrum outside the portion that is visible to the human eye. There are several examples of the application of these techniques to the output of single out-of-the-visible sensors. Although there has been research into sensor fusion, to our knowledge there has been some research into the analysis of fusion of sensor inputs either in a low-level or high-level manner in computer vision systems. However, the research that has been done has been mostly fragmented. There have been few studies in the comparison of the fusion types in the manner described in this thesis.

Low-level sensor fusion is the fusion of sensor data at the image or signal level. There are several methods for performing sensor fusion at a low level, such as various channel replacements or averaging. Many of these methods will be explored in this thesis.

High-level sensor fusion is also called decision fusion. High-level fusion is the fusion of decisions made on (or about) individual sensor signals or images. There are several methods for performing the fusion of these decisions made upon this sensor data. Like low-level fusion techniques, many of these high-level techniques will also be explored in later portions of this thesis.

C. ADVANTAGES OF SENSOR FUSION

First, a great potential exists for developing very detailed target or object signatures which can allow us to compress our incoming data to be able to detect and recognize objects with less data and better error rates. One way to accomplish this is through the use of principal components analysis, which will be covered in chapter II of this thesis.

Second, we also have the potential for overcoming some of the disadvantages of color imagery through the use of out-of-the-visible spectrum sensors. For instance, color imagery is notorious poor at areas of images that are in shadow. The use of IR, which is not profoundly affected by shadows, can overcome this disadvantage.

D. LIMITATIONS

This thesis is limited in scope to two dissimilar sensor inputs, color and thermal infrared. This thesis is only a starting point in terms of exploring methods in sensor fusion, but should serve well as a primer for further research in the field. We have also limited our research to that of detection and recognition of vehicles. Robust general detection and recognition methods do not as of yet exist in computer vision. Each method is generally tuned to a particular class of objects. Further, we have restricted our techniques to that of principal components analysis and eigenspace exploration.

The research in this thesis was conducted with a single out of the visible spectrum sensor. While many of the techniques can apply to other passive sensors, which generally return image type data, these techniques may not apply directly to active sensors, such as sonar, LIDAR and RADAR, which generally do not return simple image data. Further, the use of more than one non-color sensor could show more promise than just one.

Most non-color sensors have not been developed to the extent that color cameras have. Our IR sensor returns frames which have a much larger resolution than our color sensor. Our methods resize our IR data to the same size as our color data. However, this resizing means that there is less data contained in the IR frames than in the color frames. Because of this hardware limitation, the correspondences between color and IR data are not exact.

E. INTRODUCTION TO COMPUTER VISION AND SENSOR FUSION

In order to fully understand and develop the concepts of sensor fusion, the focus of this thesis, we must explore the basics of how images are acquired, processed and then used in recognition and detection of objects in a scene. In the systems we are considering, each sensor employed provides data to fuse, in the form of a visible image. Examples of these images are shown in Figures 1, 2 and 3 below. The next section introduces the generalities of computer vision, then we will explore object recognition and detection in terms of what we have done in this thesis.

1. Introduction to Computer Vision

Normally, in computer vision systems, detection of objects in a scene is performed first, followed by recognition of those objects. In this thesis, we first perform recognition on a very focused data set containing only vehicles in a certain pose, in order to explore the data in a more meaningful way, to understand correlations between the data sets, before performing detection. Therefore, discussion of recognition is first, followed by detection. This follows the flow of the experiments performed in this thesis.

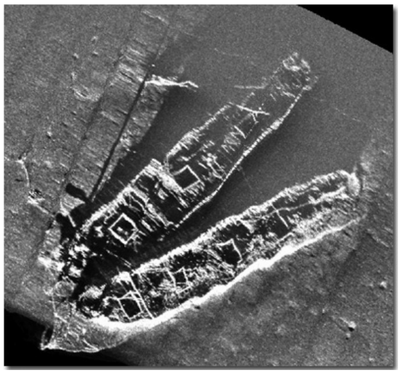


Figure 1. Sonar image of of the Frank A. Palmer and Louise B.Crary. (Courtesy of NOAA/SBNMS) .

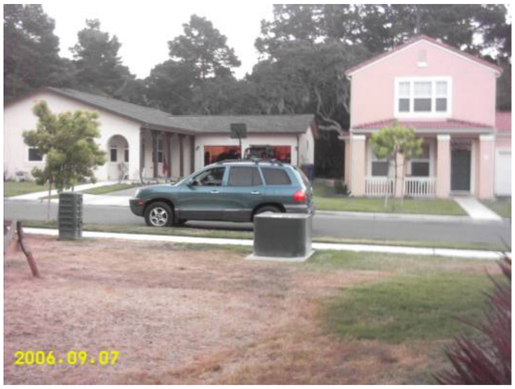


Figure 2. Color image.

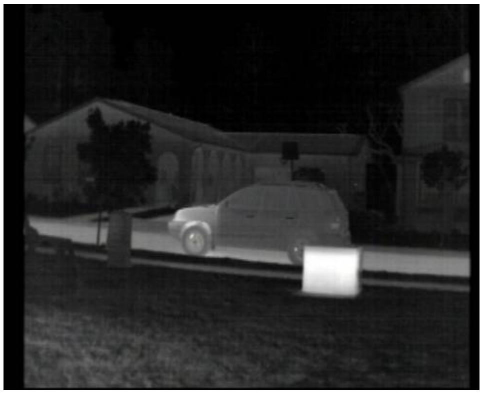


Figure 3. Thermal IR image.

a. A Model for Computer Vision Systems

Castleman (1996) put forth a model for the acquisition and use of imagery for pattern recognition. Figure 4 is an adaptation of that model. This model applies equally to both recognition and detection of objects.

According to this model, first we acquire video from our sensors and grab frames of that video. Once we have obtained our digital imagery in the form of video frames, we perform some preprocessing on the imagery, such as smoothing or some morphological operation. The intent of this first step is to smooth out and reduce the effect of noise in the image. This noise can be from the area in which the image is taken, such as trees moving in the wind, from atmospheric effects such as heat waves from a hot roadbed or from the inherent characteristics of the camera. Later stages of the computer vision system can also utilize some of these morphological operations to highlight some feature or set of features.

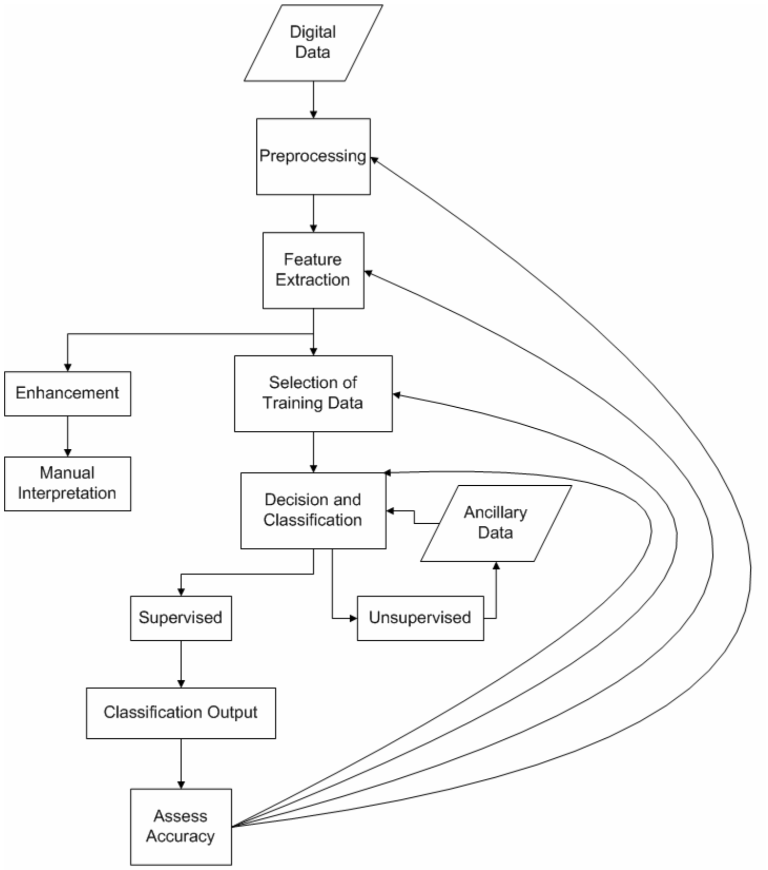


Figure 4. Idealized model for computer vision.

The second step is feature extraction. When we perform a feature extraction, we reduce the dimensionality of our data. We do this by reducing our data to only that data essential to represent the features with which we are concerned. Data dimensionality reduction is the focus of much research. Less data generally means less complexity and faster processing time, but also less information. There is a trade-off that must be considered when performing any data dimensionality reduction. This applies equally to the focusing of our imagery upon our features, explained in the next section, as well as dimensionality reduction via principle components analysis (PCA), explained in later in this chapter. The lessened complexity and increased efficiency, because of the lesser dimensionality, must be weighed by the decrease in information that is contained in the less complex data. There are methods, such as PCA, to somewhat ensure that the data that is lost is the least pertinent to the detection and recognition of objects.

In our first experiment, detailed in chapter III, we reduce our image size from 720 by 480, whole image, to 16 x 32, individual feature, in the color imagery and similarly in the infrared. This results in a feature that contains 512 pixels with eight bits of color data per channel of our image. Feature extraction is often composed of at least two steps, reducing the data dimensionality and then transforming the feature into a vector, which results in a feature vector. The feature vector is the input to the next step in the model, selection of training data.

The decision of what data to test on and what data to train on is very important. Most computer vision practitioners make a decision about what classification method to use while making a decision about what data to use for training. A poor selection can lead to over fitting. Over fitting occurs when we train a classifier on data, which is often sparse, that contains statistical irregularities that are not consistent with the population from which that data is drawn and will be used. In order for classification to be accurate in the final production system, training data must also be representative of the data that will be processed by the final production system.

In this thesis, we have used a technique called principal components analysis (PCA) (Jolliffe 2002) as a means of dimensionality reduction. In our recognition experiment, chapter III, we use PCA as a dimensionality reduction tool, then use a nearest neighbors classification algorithm. In our detection experiment, chapter IV, we use PCA and statistical measurement of test data in eigenspace to determine detection of vehicles.

We can apply this concept to a data set that has a much higher dimensionality, in the case of our first experiment 512 dimensions. If we can reduce our dimensionality by half, then we have only 256 dimensions to our data set when it is projected into the eigenspace created by PCA, greatly reducing our processing time.

Once we have our features, whether they are in image space or eigenspace, we perform classification, in our case nearest neighbors classification and statistical distance measures. After classification, we generally perform post-processing operations which often involves assessing the accuracy of our methods. Accessing the accuracy in this case really means assessing the accuracy of our feature vector classifier combination. There is unfortunately no magic feature vector, nor is there a perfect classifier for all data. The Ugly Duckling Theorem (Duda, Hart and Stork, 2001) states that there is no perfect feature vector for all data. However, there is an optimum feature vector given a certain classifier, whatever that feature vector happens to be. In addition to the Ugly Duckling Theorem is the No Free Lunch Theorem (Duda, Hart and Stork, 2001) The No Free Lunch Theorem is a corollary of the Ugly Duckling Theorem and states there is no one perfect classifier for all feature vectors.

In our case, our post-processing involved assessing how many and which eigenvectors to use in our classification. Deciding how many and which eigenvectors to retain in order to produce an optimal error rate is difficult. We use a genetic algorithm which varied the number of nearest neighbors and the number of eigenvectors to find the optimal error rate in our classification with nearest neighbors.

There are many methods and nuances of pattern classification that are not discussed here. There are several references that serve as excellent guides to pattern recognition including Pattern Classification (Duda, Hart and Stork, 2001), and Statistical Pattern Recognition (Webb, 2002). In the next section, we explore the process of detecting a particular feature vector in order to perform the recognition discussed above.

b. Object Detection in Computer Vision

This process of object detection is often split into two different processes (Sun, Bebis and Miller, 2004). First, the vision system hypothesizes where objects in the field of view could be. Then the system verifies that hypothesis by some method, often by use of an object recognition system. In our case, we perform both processes simultaneously, via distance measurement in eigenspace.

Hypothesis generation has often been done by background subtraction (Forsyth and Ponce, 2003). The process of background subtraction is one in which we develop a model of the background, generally by averaging several representative background images that do not contain any target objects. This background is then used in a form of background subtraction to determine if there are any foreground objects in any input image. Any objects in the foreground are potential objects, with which we then have a hypothesis.

In this work, we perform a type of background subtraction by extracting an area of each image. We perform this background subtraction using a threshold operation on our IR imagery to filter out only the hot areas. These areas are our generated hypotheses. Hypothesis validation can be performed by projecting the hypothesis area into eigenspace and measuring its distance from the centroid of the training set based on the Mahalanobis distance from the centroid to that projected hypothesis area. Mahalanobis distance is a variance normalized Euclidean distance measure that works well for measurement in eigenspace, because the basis of the eigenspace is determined by the variance of the training data set.

2. Introduction to Sensor Fusion

Though the concept of sensor fusion is not new, sensor fusion of dissimilar sensors is an aspect of computer vision that is beginning to receive more attention than it has in the past from many professionals and academics, in computer vision and out of computer vision. Sensor fusion means different things to different people. Some academics view sensor fusion as the fusion of individual signals from sensors, low-level, while others view sensor fusion as the fusion of data and images in geographical systems, often a high-level approach. This is due in large part to the large variety of disciplines that are working in sensor fusion and the great number of approaches to sensor fusion that one can take. For the purposes of this work, sensor fusion is defined as the fusion of data, temporally and spatially from dissimilar sensors.

The spatial fusion of images from similar sensor systems, usually called mosaicing, is a topic that has been well studied (Tso and Mather 2001) (Hall and

McMullen 2004) and often comes down to image registration from one sensor's image space to another or both sensors to some arbitrary space outside of the image space in which the sensors are employed. Because the sensor data is similar, the detection of correspondence points from one image to the next is an easier problem than that of dissimilar sensor data.

The fusion of dissimilar sensor data encompasses a set of techniques that have been in use since the first launch of the Land Sat series of multispectral satellites in 1972 (Elachi and Van Zyl, 2006). Using the output of these satellites, spectral signature patterns by which we can classify this data have been developed almost exclusively by human beings for human beings. Some effort has been made to automate some of these functions, but mostly from the standpoint of the users of these satellite systems and not towards a general purpose computer vision system.

Many scenes give computer vision systems problems. First, low light conditions, such as dusk, night and dawn, and fog and smoke, seriously hamper computer vision with color imagery alone. Similarly, areas of shadow and variable backgrounds also cause problems with computer vision and background modeling with color imagery alone. Lastly, properly constructed camouflage can easily defeat detection and recognition systems that rely solely on color imagery.

All of these problems and many more can be mitigated to some extent by the use of sensors other than color alone. Thermal IR, the non-color sensor used in the experiments for this thesis, can mitigate the problems mentioned above. Very low light conditions do not make a significant impact upon thermal IR imagery except near the two points of thermal crossover (Holst 2000). Thermal crossover is the two points during the 24 hour day in which background and foreground objects have about the same temperature. Similarly, areas of shadow in a scene do not have a large effect on IR imagery, as IR sensors senses only reflected and emitted thermal radiation and not differences in reflected light from shadows in scenes that are observed in color imagery. A variable background can also be in part mitigated by thermal IR sensors, because much of the background variability in a scene is due to differences in reflected light from moving objects, such as trees moving in the wind. With thermal IR, objects of interest are generally not affected by these changes in illumination (Kaplan 2007) (Kruse 2001).

In addition, sensors are getting much cheaper and more capable than just a few years ago. The uncooled microbolometer thermal IR sensor has only relatively recently been used in the civilian world (Razeghi and Henini, 2002). Prior to the development and release of microbolometer technology, thermal imagery of any useful quality was only obtainable from large heavy cameras that were cooled, usually by liquid nitrogen (Jha, 2000).

Note that the techniques discussed in this thesis apply mostly to passive sensors. Active sensors, such as RADAR and LIDAR, which must impart some form of energy into the environment, are also getting cheaper and more capable, but they are still prohibitively expensive and large at this point. At some point, active sensors will be cheap enough and small enough that researchers may be able to use them in a study of sensor fusion without the prohibitive cost.

There are two major approaches to dissimilar sensor fusion, high-level fusion and low-level fusion (Polani, et al, 2004) (Nett and Schemmer, 2003). These two are also called information and data fusion respectively (Kokar and Tomasik, 2001). The boundary between the two categories can be diffuse. In addition, the definitions and boundary can depend in large part on the sensor types. In the next several chapters we will describe the two categories. We will also discuss some of the methods used to fuse data with the two major sensor fusion approaches. We will then present some of the strengths and weaknesses of the two fusion types in detection and recognition of vehicles.

F. THESIS OUTLINE/ORGANIZATION

In Chapter II we review some current literature relevant to the topics in this thesis. In Chapters III and IV, we discuss our first and second experiment covering the recognition and detection of vehicles with various types of sensor fusion. Finally, in Chapter V we discuss our results and some recommendations for further work.

II. LITERATURE REVIEW

There are many approaches to and much literature concerning computer vision. There is also a considerable amount of literature exploring some aspect of sensor fusion or another. In this chapter, we explore some of this literature as it pertains to the approaches used in this thesis. In addition, we will explore literature concerning the use of IR, especially in the detection or recognition of vehicles.

A. OBJECT RECOGNITION

Much recent work in recognition in computer vision concerns facial recognition. Approaches to facial recognition are a good upper bound in terms of complexity and time for recognition of many objects with similar geography of features and relative stability among parts, given the comparative complexity of faces and many other objects. Facial recognition is germane to the recognition of vehicles, the aim of experiment 1, due to the commonality of the relationship between parts of a vehicle and parts of a face in addition to the nature of the relationship of those parts in cases of rotation and occlusion. Zhao, et al., (2003) and Yang, Kriegman and Ahuja (2002) are two comprehensive survey papers on facial recognition which surveys the use of feature based methods, template matching and appearance based methods.

Appearance based methods, in which objects in the frame are modeled based on their appearances in the frame and not on some 2D model that has been previously gathered, have been used extensively in this thesis. Among many appearance-based methods, eigenfaces have shown themselves to be of great utility. Turk and Pentland (1991) wrote a seminal paper on the use of eigenfaces for recognition. A later paper by Turk (2005) describes the use of feature based eigenface techniques in the context of the complementary use of feature based and appearance based methods.

The current use of the term recognition concerns the labeling of an image or parts of an image that have been found. One approach to recognition involves locating key features or feature points in an image. A powerful technique for accomplishing this task is SIFT (Lowe 1999). SIFT stands for scale invariant feature transform. This technique and follow-on techniques based upon it are based on finding local scale invariant key points in images. Principle Component Analysis (PCA) (Duda, Hart and Stork 2001), is a different technique in which the practitioner attempts to reduce the dimensionality of the data in the recognition phase of a computer vision system. By representing the data in less dimensions, classification can be quicker in comparison to the full data set although with potential loss of information.

In this thesis, we have used a technique called principal components analysis (PCA) (Jolliffe 2002) as both a means of dimensionality reduction. In our recognition experiment, chapter IV, we use PCA as a dimensionality reduction tool, then use a nearest neighbors classification algorithm. In our detection experiment, chapter V, we use PCA and statistical measurement of test data in eigenspace to determine detection of vehicles.

PCA is a method in which we re-project our data into eigenspace, which has as it's basis, a set of orthonormal eigenvectors which account for the axes of most variance our data set. The first eigenvector is projected in image space as a straight line that uses the centroid of the data set as its zero point origin and lies in the direction of the most variance. The second eigenvector lies in a direction orthogonal, to the first eigenvector and takes into account the next largest direction of variation in the data set. The amount of variation in the data set is expressed in the eigenvalue that is associated with each eigenvector. An illustration of PCA, of a very simple data set consisting of a set of two-pixel images, is shown in Figure 5 below.

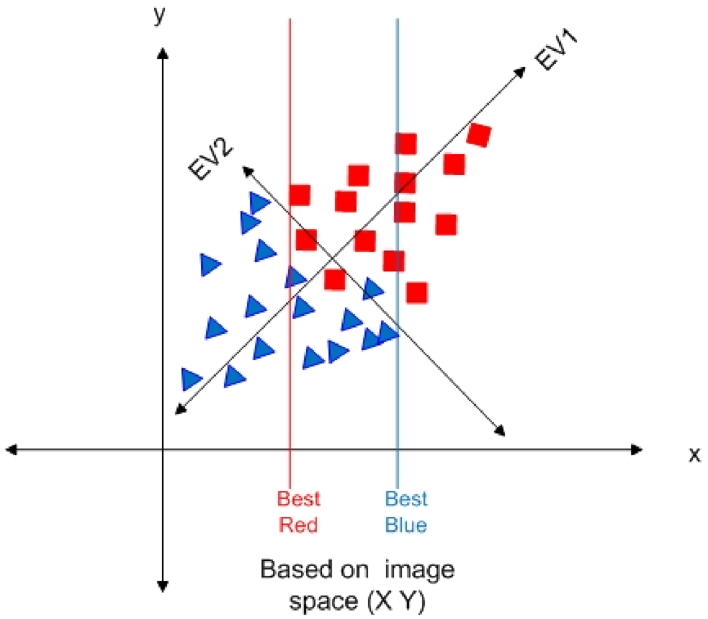


Figure 5. PCA on a simple two pixel image set.

In this illustration, we can see that in the original image space, the X-Y plane, the best decision boundary in relation to either red or blue objects captures many images in the opposite class. This would result in very poor performance in our classifier. Instead, if we create an eigenspace, based on a PCA performed on the image set, we get the two axes, EV1 and EV2. This is where dimensionality reduction is done in PCA. Based solely upon the point the data set is projected upon on EV1, we can create a decision boundary as shown in Figure 6.

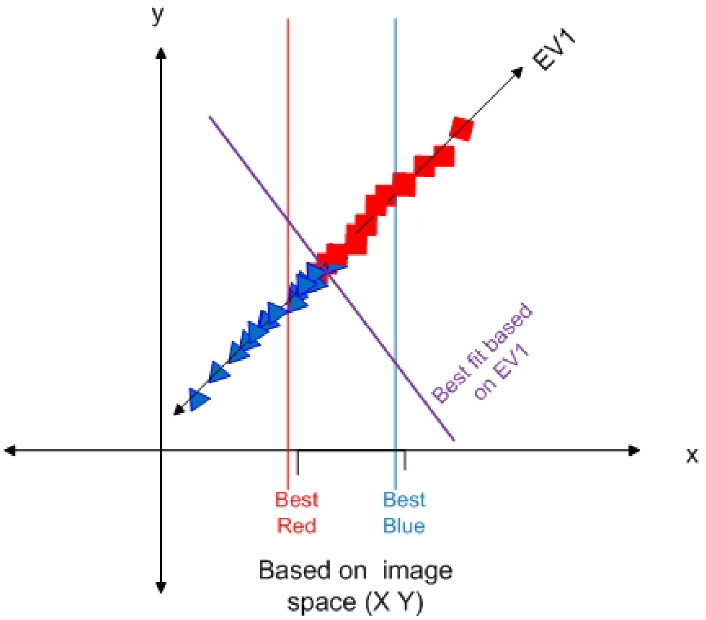


Figure 6. Decision boundaries based solely upon data projection onto EV1.

When we represent our data set solely with each image's position on the EV1 axis we can draw the linear discriminant on the best decision boundary based on that axis as shown in Figure 6 with a lessened error rate. Further, this results in a reduction of the dimensionality of the data required to produce an optimal linear decision boundary. Therefore, we have a reduction by half in our data dimensionality with no corresponding increase in error rate associated with our data set. Note that this is a contrived example. Real data sets are rarely this clear, but the principles apply none the less.

Images can be projected into and out of eigenspace. Projecting an image into eigenspace with all associated eigenvectors results in no loss of data. We can reduce the dimensionality of our data in eigenspace by using less than the full number of eigenvectors to project our data into eigenspace, though we will lose information. This lost data is generally data that is of the least use to our classification because the eigenvectors thrown away are those with the least variation. Reconstruction of images from eigenspace is possible but is generally only performed to observe what pixels are responsibly for the most variation. Further discussion of image projection is reserved for Chapter III and IV.

B. OBJECT DETECTION

Object detection is often done before object recognition (Brown, 2004). As mentioned in Chapter I, object detection typically begins with the generation of hypotheses for the location of an object of interest. Hypothesis generation can be classified into three different categories (Sun, Bebis and Miller, 2004):

- Knowledge based
- Stereo based
- Motion based

Knowledge based methods make use of a priori knowledge concerning target objects such as color (Buluswar and Draper, 1998), texture, symmetry (Kuehnle, 1991) and edges (Betke, Haritaglu, and Davis, 2000). Stereo-based methods often use either disparity maps of differences in object location from images in two different locations or antiperspective transformation. An antiperspective transformation is an inverse transformation of data from one sensor’s field of view to another’s using inverse perspective mapping. Motion based methods hypothesize the location of objects based on optical flow (Giachetti, Campani and Torre, 1998). Optical flow methods work well for moving objects. However, detection of vehicles, which are standing still, is not practical using this method, unless the camera is moving in relation to the still vehicle. In experiment two, we use a knowledge-based method in which we compare a random sub-image of a frame with the eigen-dataset developed from prior training data.

Although not specifically mentioned in Bebis, Sun and Miller, background subtraction (Brown, 2004) is another method frequently used for hypothesis generation. Background subtraction compares the current image with a base image or set of images. The difference between the two is used to determine what in the image is background and what is foreground. The foreground contains object location hypotheses. However, background subtraction can be complicated, and almost made impossible, by complex outdoor environments, interactions between moving object and stationary or moving backgrounds or even variations in lighting conditions over the time of sequence of frames (Brown, 2004).

As we can see, there are many approaches to object detection. The approaches mentioned here are approaches that have at least in part contributed to this thesis.

C. USES OF INFRARED IN COMPUTER VISION

Infrared imagery has been studied for some time in computer vision. The use of infrared imagery is often for vehicle detection (Kagesawa, 2001) (Nelson, 2001) (Der and Chellappa, 1997). There have been some unique uses of infrared imagery in computer vision. Der and Chellapa (1997) use a probe based method, which uses a form of background modeling to recognize vehicles in single forward looking infrared (FLIR) images. In the case of Der and Chellapa, a probe is the output of a Probe-Based Automatic Target Recognition in Infrared Imagery simple mathematical formula that operates on a pixel and it's neighboring pixels. These values are used to determine the probability that a certain shape target is present. Kagesawa, et al. (2001) use infrared to extract local features of vehicles to make vehicle recognition invariant to the many variations in vehicle color. They use two different methods with varying success. First, they use eigenvectors, which in their implementation are accurate but slow. Then they use a vector-quantization method, which is fast but inaccurate. Because Kagesawa's eigenvector methods are germane to our work, we will explain them in more detail.

To start with, Kagesawa et al compute image "windows" based on a standard corner detection algorithm. These windows are local features of the vehicles in their training set which they use to create a set of eigenvectors. Kagasawa then uses this eigenvector set to project similarly selected features from their test set into eigenspace. These projected features are then used with a nearest neighbors algorithm to decide to which class the window belongs. Using this method, Kagasawa receives detection and recognition rates of greater than 90%.

Detection of armored vehicles in IR images is also a topic of current interest. Nelson (2001) uses infrared imagery with a fuzzy inference-based detection and classification system, somewhat related to a cascaded classifier, to detect and classify tanks and armored vehicles. Andreone et al (2002) develop an approach for the detection and tracking of vehicles initially using hot areas of the image only. These areas are then refined using domain knowledge concerning size and shape of vehicles. This method proved effective at a range of 20-100 meters at 12 frames per second. This is similar to an approach we use in experiment two.

D. SENSOR FUSION

There are many instances, from many disciplines, of the use of multiple sensors, especially those of remote sensing and geographic information systems (GIS) (Nandhakumar 1991) (Tso and Mather 2001) (Hall and McMullen 2004). Until recently, the use of multiple sensors referred, depending in part on the discipline, mostly to the use of multiple visible light cameras (Yilmaz 2007). The use of multiple sensors in geographic information systems and satellite imagery is well documented (Tso and Mather 2001). Practitioners use multiple sensors in GIS to create an image that contains patterns designed for interpretation by human beings. The use of dissimilar sensors in sensor fusion in an autonomous way is part of the forefront of computer vision. The use of sensor fusion in computer vision systems can be broken up into two major categories, high level and low level. An example of high-level fusion is the use of sonar and laser range finders to locate people with robots (Martin et al 2005) in which each sensor maintains its own Gaussian probability distribution of the belief that a human being is in the field of view. The evidence from these multiple sensors is then combined in a probabilistic aggregation scheme, similar to the voting scheme described in chapter IV. Another example of high-level fusion is by Hunke and Waibel (1994), in which they develop detectors for multiple attributes of human beings, such as color, shape and movement. They feed each detector with the results of the previous detectors. Cramer, Scheunert and Wanielik (2003) compare the use of both categories in the fusion of LIDAR and infrared sensor data in detection and tracking using a parts-based approach. In this case, the parts are the parts of a pedestrian, such as the legs and arms, which are modeled because they move in a periodic motion relative to the pedestrians trunk.

Low-level fusion approaches are more plentiful. Piella (2002) approaches this problem with a region-based approach that can deal with multiple resolutions using segmentation of imagery at multiple resolutions to find structures that can be correlated between images. Mitianoudis and Stathaki (2007) use a similar approach with the inclusion of a pixel based approach in PCA. They perform PCA on several patches of each image they intend to fuse. Then they keep the N best bases from that PCA to use in ICA, Independent Components Analysis. They use an estimate of the ICA bases to do an ICA transform to fuse the images.

Brown (1992) is an excellent survey of image registration techniques many of which are still valid today, and which can greatly ease the low-level fusion of sensor data.

E. SUMMARY

In this chapter, we discussed several approaches and pieces of literature to fill in the holes that could not be filled in the introduction to computer vision and sensor fusion discussed in Chapter I. Further, these approaches contribute to our approaches to detection and recognition, Chapters III and IV respectively, in many ways.

III. RECOGNITION EXPERIMENT

A. INTRODUCTION

The first experiment we conduct in this thesis concerns the recognition of vehicle classes in color and IR imagery, via classification of fused and raw sensor inputs. Generally, in computer vision, object detection is done before object recognition. However, we have reversed the process in order to conduct a more detailed study of the feasibility of fusion in terms of objects, the nature of sensor data input and target/object signatures. This study of the fusion of smaller areas of images was essential to the fusion of whole images, which we use in detection, detailed in Chapter IV.

The purpose of this experiment is to determine the best classification error rate that we can achieve by the use of various low-level fusion techniques as well as using each sensor's individual input without fusion. The classes of our vehicles are cars and trucks/SUVs/vans. Examples of each class, in color, are shown in Figures 7 and 8.

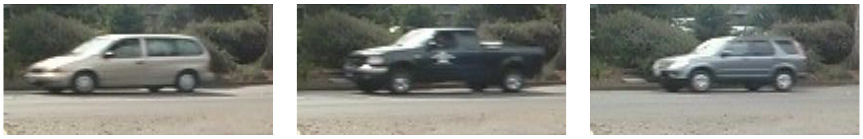


Figure 7. Representative members of the truck/SUV/van class.

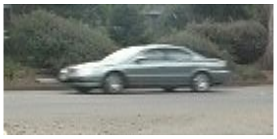


Figure 8. Representative member of the car class.

Our data was captured in daylight, between 12:00 and 12:30pm, approximately 40 feet from a road on the campus of the Naval Postgraduate School. A full size color image, captured at 320 x 240, is shown in Figure 9. A full size IR image, captured at 320 x 240 pixels and then upsampled to 640 x 480, is shown in Figure 10. Vehicles were traveling laterally at less than 30 miles per hour. The background was variable with variable winds and shadows as well as varying levels of sunlight due to moving cloud cover.

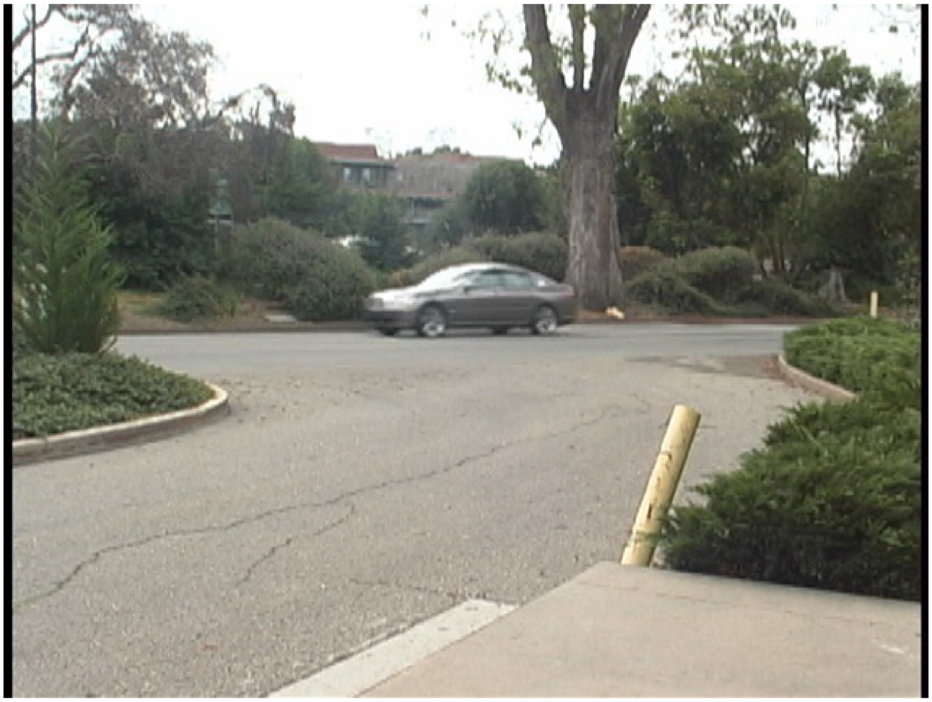


Figure 9. Full-size image of experiment location and view in color.

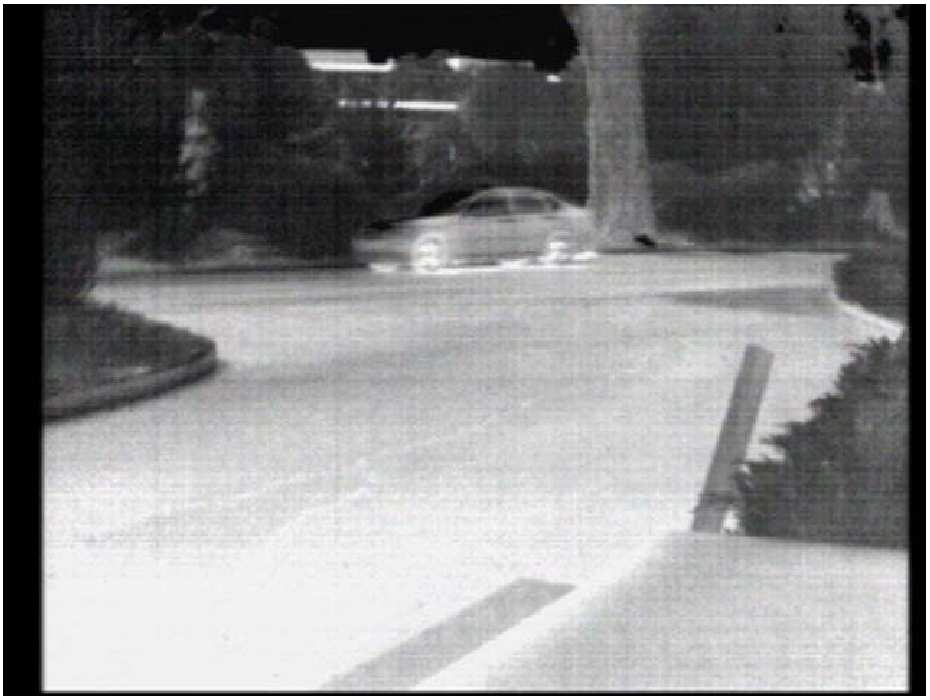


Figure 10. Full-size image of experiment location and view in IR.

Our cameras were collocated approximately 6 feet from the ground, with 6 inches from camera center to camera center as shown in Figure 11, pointed in the same direction and zoomed to include approximately the same scene area, as shown in Figures 9 and 10.

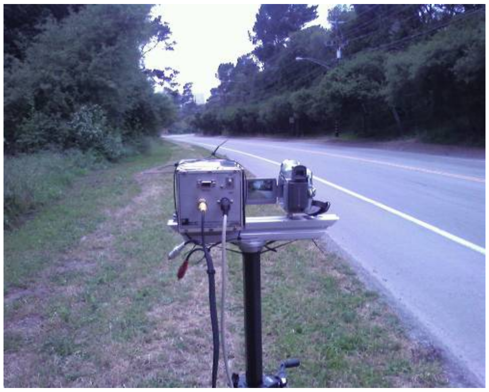


Figure 11. View of tripod and camera setup.

In this first experiment, we use principal components analysis (PCA) to reduce the dimensionality of our training image set. In our case, the principal components are linear combinations of raw pixel values, represented by eigenvectors. Once we determine the eigenvectors, we project images into eigenspace and use the k-nearest neighbor classification scheme (Duda, Hart and Stork 2001) to determine to which of two vehicle classes the projected image belongs. Our training/testing validation method is “leave one out.”(Mitchell 1997) This results in 50 training images and 1 testing image for every iteration of the training cycle.

B. SPATIAL AND TEMPORAL REGISTRATION

Spatial and temporal registration is a problem that is of great importance in both types of fusion, but much more so for low-level fusion. This problem can be greatly diminished by acquiring a sensor suite that is setup to be temporally synchronized. In our case, we define temporal synchronization as developing a stream of frames from two or more sensors such that each frame in each sensor data stream temporally corresponds, within some tolerance, to every other frame taken at approximately the same time from every other sensor in the given sensor suite. Several manufacturers have created dissimilar sensor suites that are temporally synchronized, though these sensor suites are often limited to two sensors, one of which is color. Temporal registration, however, is the easier problem to address. Even if we do not have a synchronized rig, we can work out our synchronization such that the effects from temporal differences is greatly minimized, depending greatly on the context of our objects of interest. In the case of slower moving objects and sensors that have the same frame rate, the problem is quite simple to solve using rudimentary methods.

In our experiments, this chapter and Chapter V, we temporally synchronize by utilizing a ground reference point in both sensor videos. The sensor point is a tree in the middle of the field of view in the first experiment. The large tree is shown just to the right of center in color in Figure 12 and in IR in Figure 13. We clipped the videos using commercial video editing software so that both videos started when the same vehicle reached the extreme edge of the tree, found by zooming in to the tree and clipping the videos when the edge pixels of both cars meet the edge pixels of the tree. Using this technique, we achieve accurate temporal registration between the videos and very accurate color to IR frame registration as described below.

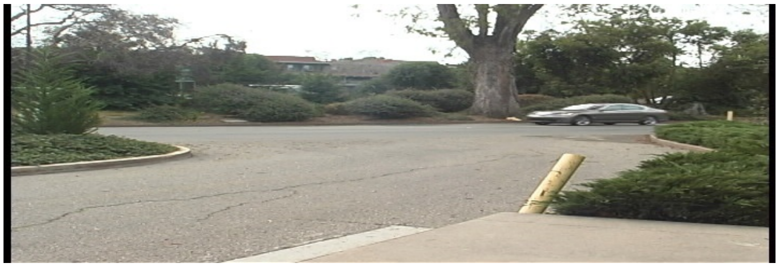


Figure 12. Example color image used for temporal registration.

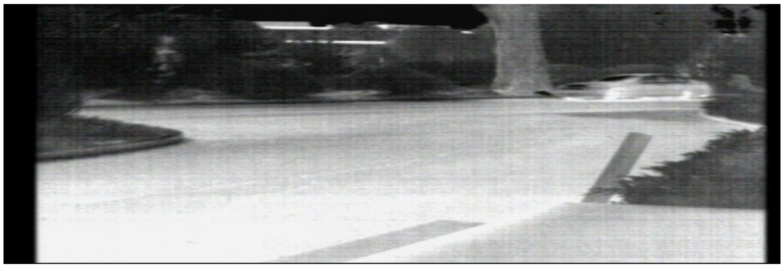


Figure 13. Example IR image used for temporal registration.

To show the accuracy of registration, we note that in each video, our output is set to 720 x 480 pixels before deinterlacing. The field of view at the closest point in which we have vehicles, about 40 feet away, is 612 feet horizontally. At 720 pixels wide, we get:

$$\frac{90\,feet}{720\,pixels} = 0.125\ {}^{feet}\!\big/_{pixel},$$

approximately, on the track of the vehicles in the video. If we assume, as a worst-case argument, that the vehicles are moving at 45 miles per hour directly lateral to the cameras, then we get:

$$\frac{45 miles}{1 hour} * \frac{1 hour}{3600 second} * \frac{5280\,feet}{1 mile} = 66\ {}^{feet}\!\big/_{second}$$

Each frame is 1/30 second of video. Therefore, from one frame to the next we have:

$$\frac{66\,feet}{second} * \frac{1 second}{30} = 2.2\,feet$$

of movement. 2.2 feet of movement at that distance from the camera translates to:

$$2.2\,feet * \frac{8\,pixels}{foot} = 17.6\,pixels$$

of movement between frames. Therefore, even if we missed our mark by the worst case scenario of nine pixels, we would still have temporal registration with an absolute real difference that is less than the difference between any two frames, given the lateral speed of the vehicles of interest.

Spatial registration is the other synchronization problem that must be addressed, especially in the case of low-level fusion. Different sensors represent data and objects in often very different ways. Sizes and shapes of the same objects can be completely different, even with the same focal lengths and zoom levels between two dissimilar sensors. As illustrated in the figures below, because of the heat signature from the vehicles wheels and engine, the size of the vehicle in the two images is completely different, even though the zoom levels and focal length are similar between the two sensors.

In our case, we perform spatial recognition using known point correspondences in both images. We use these known point correspondences to compute a homography matrix. We then use this homography matrix to transform each pixel of the IR imagery into the image space coordinates of the color sensor. This results in each pixel in IR that contains data being in direct relation to the pixel with the same coordinates in the color frame, since the frames have been previously temporally synchronized.

It is important to note that temporal synchronization is best done in hardware. Using hardware, we can perform a genlock that ensures that as one sensor takes a frame, the next sensor also takes one (within some temporal tolerance). While spatial synchronization can also be done in hardware, it is trivial to perform the synchronization in software.

C. METHODOLOGY

After setting up and getting our data videos from each sensor, we process each video to edit out portions of the video without vehicles to get the most useful set of images. We grab image frames from the video, which are recorded at 30 frames per second. Both our cameras record video interlaced. The cameras capture an image and read out the image one row at a time, skipping one row before every scan. The rows that were skipped are filled in by the next field. This means that as the image data is scanned any moving object in the scene results in jagged edges of every other line because of the difference in position between scan times, as show in Figure 16. We used only the even fields, resulting in the images shown in Figures 14 and 15, which appear compressed in the vertical direction, as opposed to the full height, still interlaced, image in Figures 9 and 10.

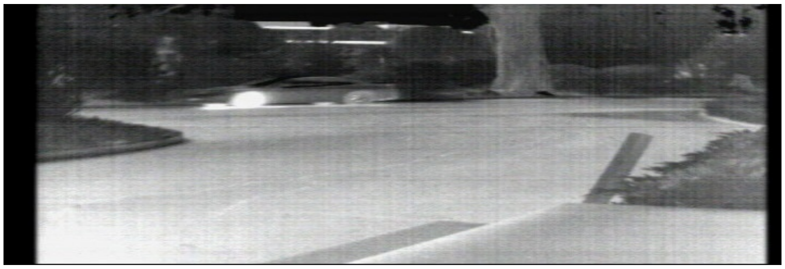


Figure 14. Radical appearance difference: IR.

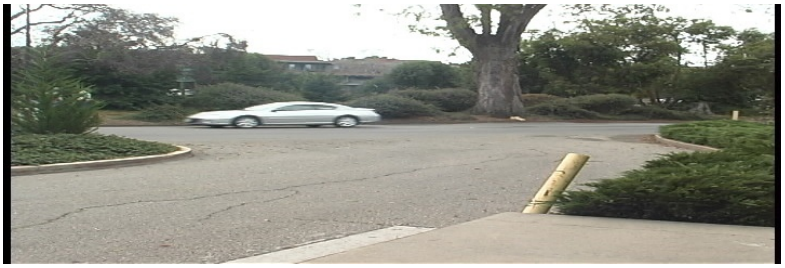


Figure 15. Radical appearance difference: color.

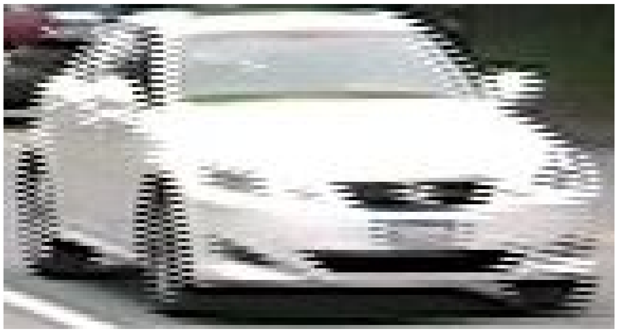


Figure 16. Illustration of interlacing artifacts.

Once our images are deinterlaced, we further restrict our images to the set of vehicles that drove from right to left in order to develop a homogenous training data set, by which we can more make meaningful and tangible comparisons.

We then grab features using a constant aspect ratio region of interest (ROI), with a ratio of width to height of about 1.400. We use the constant aspect ratio ROI because it lessens the effects of vehicles that are at times quite different in size. The ROI is drawn to include the edges of the front and rear bumper and the bottom edges of the vehicle’s tires.

Before converting our features into feature vectors, we preprocess the ROI with a 10x5 pixel Gaussian kernel. We used a rectangular kernel due in large part to the rectangular nature of our known objects. We then sub-sample the ROI to resize them to 16x32 pixels. The blur is performed to reduce the effect of aliasing which is common when sub-sampling images. The resizing is performed in order to, first, make every feature a common size and, second, to lessen the amount of data we have to process, while minimizing the loss of essential data, such as some of the data around the vehicles.

This processed feature data is vectorized and combined together with all of our training image data to form an image stack. This image stack is an m x n matrix in which m is the size of the vectorized ROI and n is the number of images in our training set as illustrated in Figure 17 below. In our case, m is 16 x 32 = 512 elements for grayscale imagery and 16 x 32 x 3 = 1536 elements for RGB imagery.

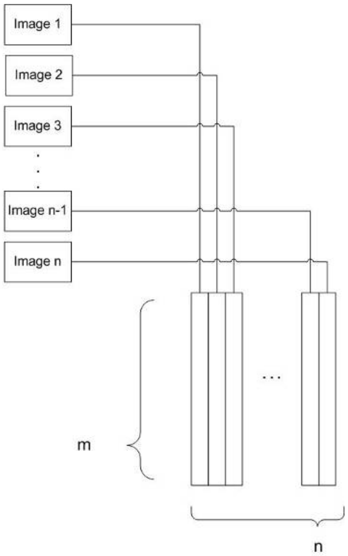


Figure 17. Creation of an m x n image stack.

Once we have our image stack, we process the image stack in feature space via PCA to transform the images into eigenspace. First, we generate the mean image from all the images in the image stack. Then, we subtract the mean image from each of the images in the image stack, so that each row, a, in A is:

$$\begin{aligned} a_1 &= image_1 - \mu_A \\ a_2 &= image_2 - \mu_A \\ &\vdots \\ a_n &= image_n - \mu_A \end{aligned}$$

With the image stack in this form, we compute the covariance matrix associated with all of the training images by multiplying the transpose of the image stack by the image stack itself.

$$COV = A^T * A$$

After we create the covariance matrix, we use it to calculate the eigenvectors and eigenvalues associated with the image stack and sort them according to the variance along each of the eigenvectors. The eigenvalues are the variances associated with their respective eigenvectors.

To reduce the number of eigenvectors used to project our test images into eigenspace, we use a “semi-diagonal” square matrix, the size of our eigenvalue matrix, in which we place 1’s in the diagonal for any eigenvector we wish to explore and zeros all other places. We multiply this matrix by the matrix of eigenvectors, as shown below to filter out all but the first n , in this case n=3, eigenvectors:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} ev_{11} & ev_{21} & ev_{31} & ev_{41} & \cdots & ev_{m1} \\ ev_{12} & ev_{22} & ev_{32} & ev_{42} & \cdots & ev_{m2} \\ ev_{13} & ev_{23} & ev_{33} & ev_{43} & \cdots & ev_{m3} \\ ev_{14} & ev_{24} & ev_{34} & ev_{44} & \cdots & ev_{m4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ ev_{1n} & ev_{2n} & ev_{3n} & ev_{4n} & \cdots & ev_{mn} \end{pmatrix} = \begin{pmatrix} ev_{11} & ev_{21} & ev_{31} & 0 & \cdots & 0 \\ ev_{12} & ev_{22} & ev_{32} & 0 & \cdots & 0 \\ ev_{13} & ev_{23} & ev_{33} & 0 & \cdots & 0 \\ ev_{14} & ev_{24} & ev_{34} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ ev_{1n} & ev_{2n} & ev_{3n} & 0 & \cdots & 0 \end{pmatrix}$$

For example, for our grayscale imagery our eigenvector matrix is 536 x 51 and our eigenvalues matrix is 51 x 51. Normally we would expect the eigenvector matrix to be square. However, we used a trick from Turk and Pentland (1991) which results in a covariance matrix that is (number of examples) x (number of examples). This matrix is much smaller than the standard covariance matrix and therefore results in much faster computation of the eigenvectors of the covariance matrix. This smaller covariance matrix is then used to create the resulting eigenvector matrix which is (number of dimensions/pixels) x (number of examples). This matrix is not square but is still an exact representation of the first n eigenvectors, where n is the number of examples.

To explore the effects of the first 30 eigenvectors we create a matrix with zeros everywhere except the first 30 diagonal places in the matrix. We then place 1’s in the first 30 diagonals. We then multiply this 51x51 “semi-diagonal” matrix with the matrix that contains the eigenvectors. This in effect filters out the last 506 eigenvectors. These eigenvectors are the eigenvectors that are responsible for the 506 smallest amounts of variance as expressed in the eigenvalues associated with those eigenvectors

Once we filter out all eigenvectors except our eigenvectors of interest, we can recreate the original images to obtain a visual explanation of the pixels of the image that represent the greatest variance in the image set. Examples of this reconstruction are shown in Figures 18-20. Figure 19 and 20 are reconstructions using 1 and 10 eigenvectors respectively. Figure 18 is the original image.

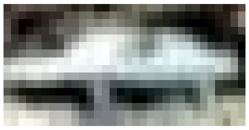


Figure 18. Original image, fused with one channel replacement.

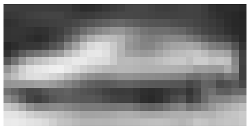


Figure 19. Reconstruction using the mean and 1st EV.

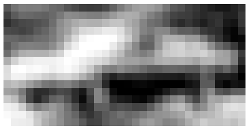


Figure 20. Reconstruction using the mean and first 10 EVs.

We use the images projected into eigenspace, varying the number of eigenvectors, to classify our test features in a k nearest neighbors classification scheme. We recreate this process for:

1. Color alone

2. IR alone

3. Fusion by replacement of the V, value or brightness, channel of the color RGB image after conversion to HSV

4. Fusion by averaging the V channel of the color HSV image and the single channel grayscale IR image.

Results and discussion of this experiment follow.

D. RESULTS OF THE EXPERIMENT

Initially we experimented with various eigenvector counts and values of k in k-nearest neighbors classification, varying the counts and values by hand. The results we received looked promising. The best classification rate (decision between vehicle classes) was as high as a 90%.

The graphs in Figures 21 through 23 below show the curve of variance captured compared to number of eigenvectors, corresponding to the eigenvector set used in the results section above, for color, V channel replacement and IR respectively. In an ideal world, the greater the amount of variance captured means the greater amount of information best responsible for classification has been captured. Assuming this to be the case, our results make sense in that the curve for color shows a much greater rate of variance capture than either IR or V channel replacement of HSV color. The variance captured for IR alone is considerably less than that for color. Consequently, the curve for V channel replacement shows that the curve is depressed in relationship to color alone, but is better than IR alone.

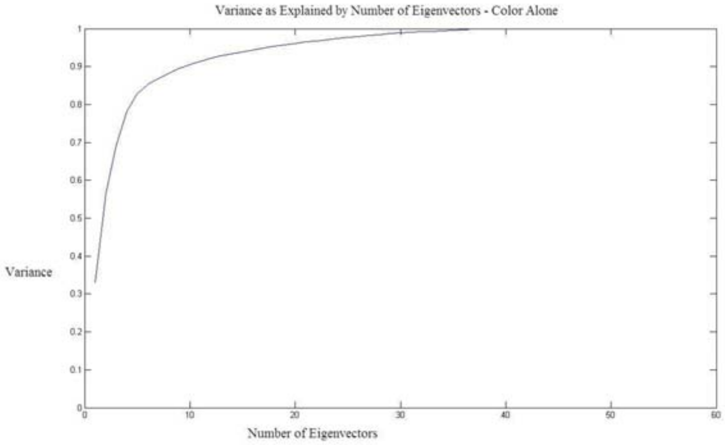


Figure 21. Variance curve color alone.

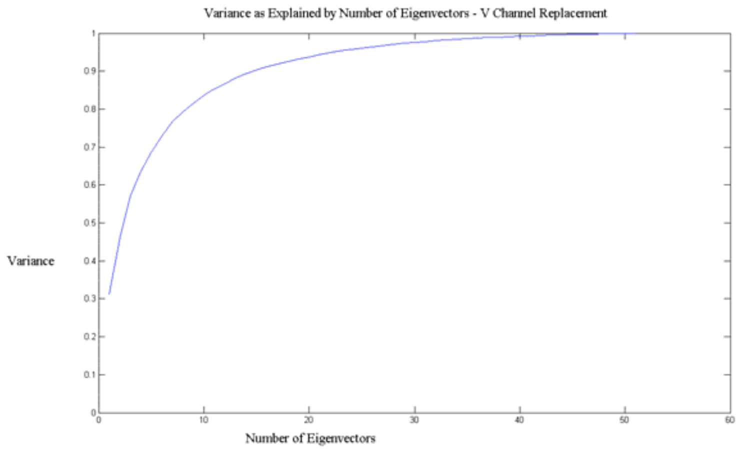


Figure 22. Variance curve v channel replacement.

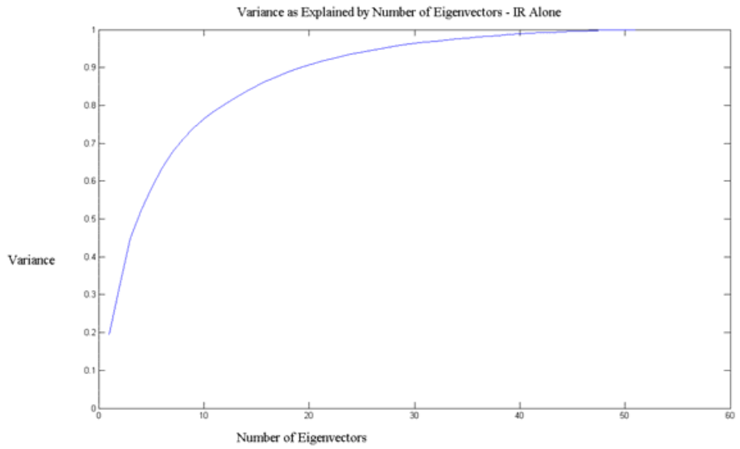


Figure 23. Variance curve IR alone.

In order to develop a more complete understanding of the quality/potential of our approach, we ran our methods varying through all eigenvectors in the set of eigenvectors, checking for best classification rates against all nearest neighbors from one to fifteen. We chose 14 as the maximum number of nearest neighbors, because we have only 14 members of the SUV/Truck/Van class. The addition of a number of nearest neighbors greater than the number of the smallest class could result in artificially depressed classification rates.

We graphed our best classification rates over the number of retained eigenvectors given every aforementioned number of nearest neighbors. The graph of best classification rate in color, IR and simple channel replacement fusion is shown in Figures 24 through 26 below.

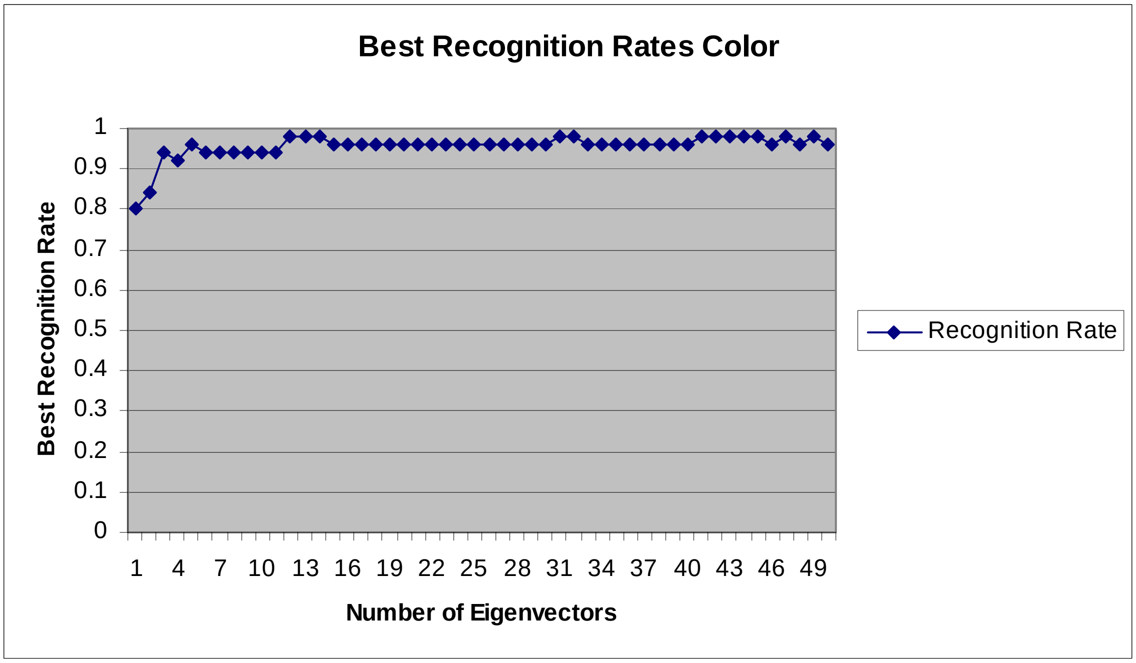


Figure 24. Best recognition rate using color.

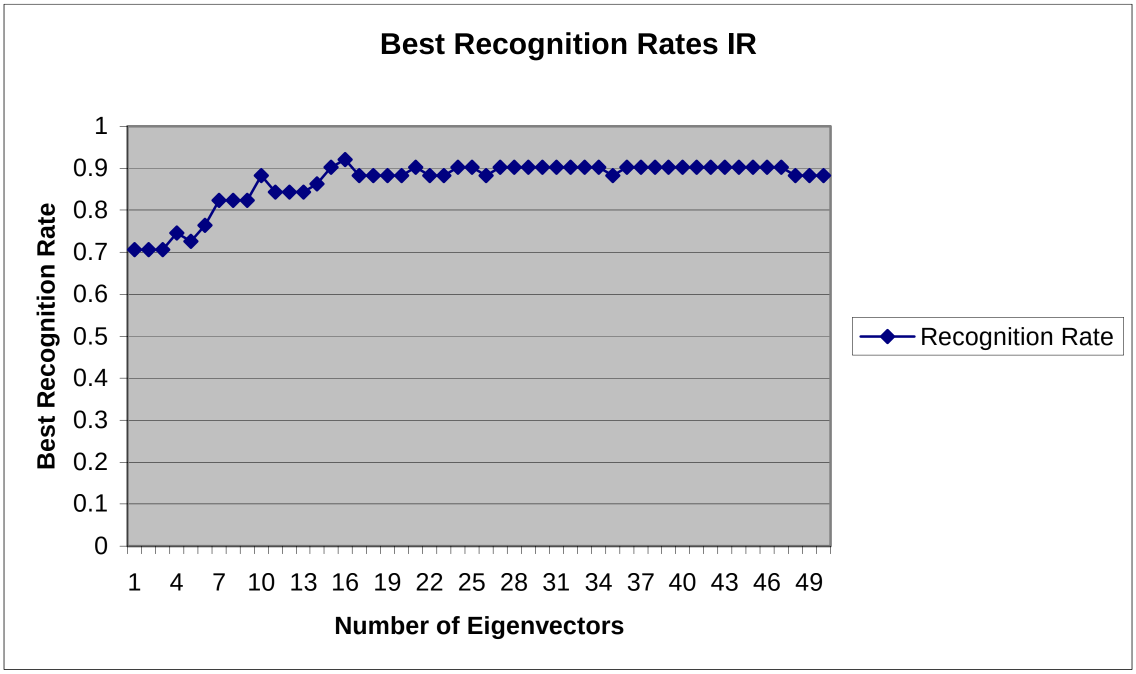


Figure 25. Best recognition rate using IR.

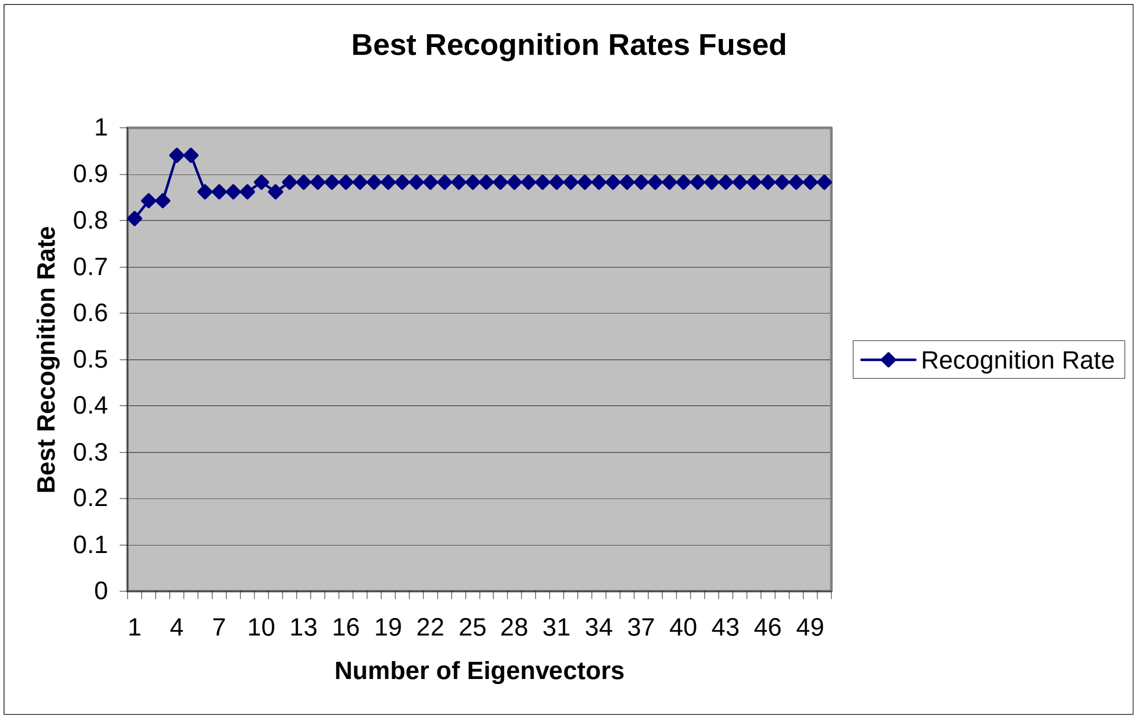


Figure 26. Best recognition rate using fused imagery.

Clearly, the color recognition rate is overall superior to both the IR and fused rates. The rates of recognition for color and fused imagery are comparable, with color no more than 2 percent better, given less than five eigenvectors. The data which these graph were made are contained in Appendix A.

E. DISCUSSION

Based on Figures 23 and 25 and the results shown in the last section, thermal IR is not a good sensor to use for recognition of vehicle classes. We hypothesize that this is due in large part to the variation in heating of vehicles due to the distance the vehicles have driven, the type and tread of tires, which create differential friction with the roadbed and variations in body heating due to the position of the vehicle in relationship to direct sunlight. Figure 27 below shows two examples of the variation in heating of vehicles in the car class that make the vehicles appear very different. Figure 28 is an example of the truck class in IR. The heat signature of the truck is not different enough from the two cars to make classification easy.

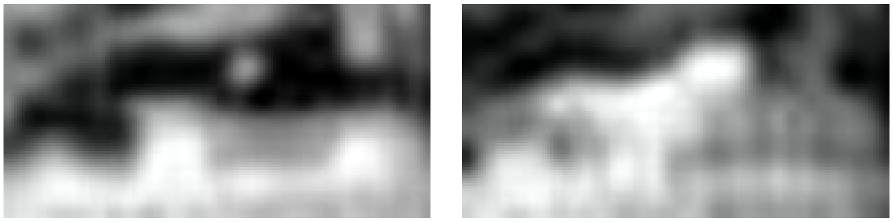


Figure 27. Members of the car class in IR.

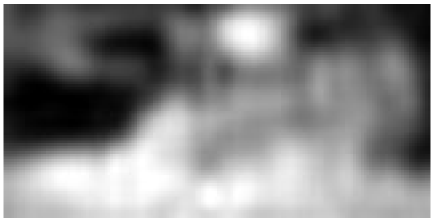


Figure 28. Member of the truck class in IR.

Clearly from the graphs shown in the results section above along with the graphs of the variance captured by our sets of eigenvectors, color is preferable to either IR or fused data for recognition of vehicles using the methods employed in this thesis. This does not mean that there is not a better means to classify using this data that could make better use of the fused data; it only means that our classifier is more optimal for color data versus IR or fused data.

F. SUMMARY

Given our data and our classifier/feature vector combination, color data is superior to either fused or IR data. The next experiment, detection using fusion, contained in Chapter IV, explores more types of fusion than the simple fusion shown in this chapter for the detection of vehicles. In Chapter V, we present our overall results and present some follow on work that could result from this thesis.

IV. DETECTION EXPERIMENT

A. INTRODUCTION

The second experiment we conduct in this thesis concerns the detection of vehicles in color and IR imagery, via classification of fused and raw sensor inputs.

This experiment differs from the first experiment in several very important ways. We pose detection as a classification problem where the two classes are “vehicle” and “background/other.” There is a lot of variation within the vehicle class, but there is a much greater variation in the class of all objects not a vehicle. Second, we have a huge search area. In any given image, we first do not know if we have a vehicle or not. Even if we have an image which we know contains a vehicle, the vehicle in our imagery is about 150 x 50, 750 pixels, at the largest. Each image is 720 x 240 or 172,800 pixels. Using simple template matching scanning every 150 x 50 region, that leaves over 170,000 regions we must sift through before we find, if we find, our vehicle.

The purpose of this experiment is two fold. First, we will explore the ways that dissimilar sensors can be fused in a high-level manner to focus our search. Specifically, we will explore the ways that IR can be used to focus the search for vehicles in color. Second, we will determine the best detection rate that we can achieve by the use of various low-level fusion techniques as well as using each sensor’s individual input without fusion.

B. DATA COLLECTION

For this experiment, our data was captured in the early morning light, between 6:30am and 7:30am on March 15, 2007. Our tripod was approximately 5 feet west of Aquajito Road, 100 feet south of Farragut Road. Figure 29 below shows the tripod setup. The temperature was 52° F. Our cameras were 6.5 inches from optical center to optical center, 4 feet 5 inches from the ground. The speed limit on this road is 35 miles per hour.

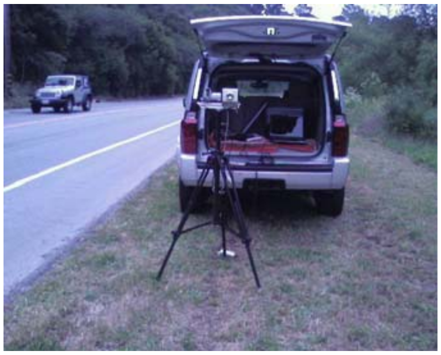


Figure 29. Tripod set up on Aquajito Road.

Due to the low temperature and the nearness to the point of thermal crossover, our IR imagery is not of the best quality. The upside is that any techniques that work well in these conditions will probably work better given better conditions. An example IR frame is shown in Figure 30 below. For comparison, a color frame is included in Figure 31 below the IR frame.

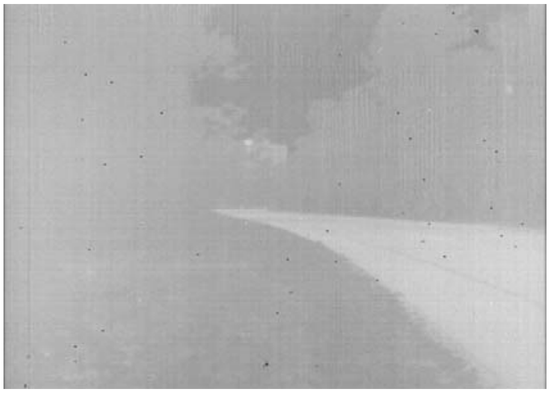


Figure 30. IR frame example.

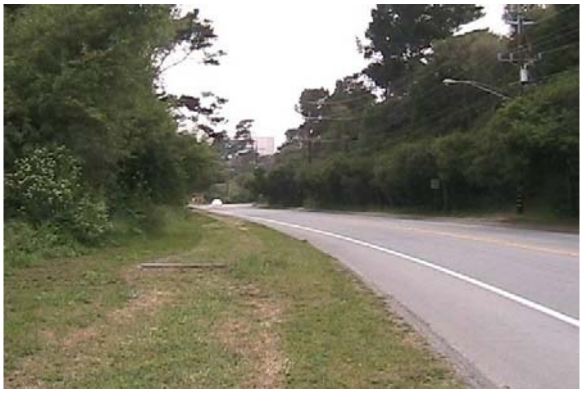


Figure 31. Color frame example.

C. DETECTION USING HIGH-LEVEL FUSION

1. Introduction

High-level fusion is sometimes called decision fusion. In high-level fusion, each sensor in a sensor suite creates an image, from which a unique feature vector is created which is then used as the input into a classifier created for that sensor's data. In a high-level fusion approach, for example, we can treat an RGB color image as three separate inputs, the R G and B channels, and create a feature vector based on each channel, then use each of those feature vectors as input into one classifier for each channel to make a decision concerning the presence and type of an object in a scene. With this kind of approach to sensor fusion, we fuse the decisions of the several different classifier/feature vector combinations in order to make a decision concerning the input. In this way, the preponderance of the evidence may point to one object or another, in such a way that we may not be able to see in low-level fusion. Example approaches to high-level fusion include (Hall and McMullen 2004):

- Cascaded classifier – In this approach to fusion, we string out our classifiers in such a way that the less expensive classifications are done earlier. The fusion system can then make a decision, based on the classifiers, that the cost of advancing to the next stage is more costly than making a decision given the stages already completed. If this system decision has a degree of certainty that is acceptable, then it should process no further. A cascade classifier is illustrated in Figure 32 below.

Advantage: This approach can be less computationally expensive, because further levels of a particular cascade may not be used very often.

Disadvantages: Optimal setup of the classifier stages is very object-sensor suite specific. We must understand what sensor and feature vector is best for our object of concern, as that is the only way to ensure that we do not use the more complex stages before a less complex and more desirable discriminator. Early errors propagated through the classifier cannot be easily found or fixed in later stages.

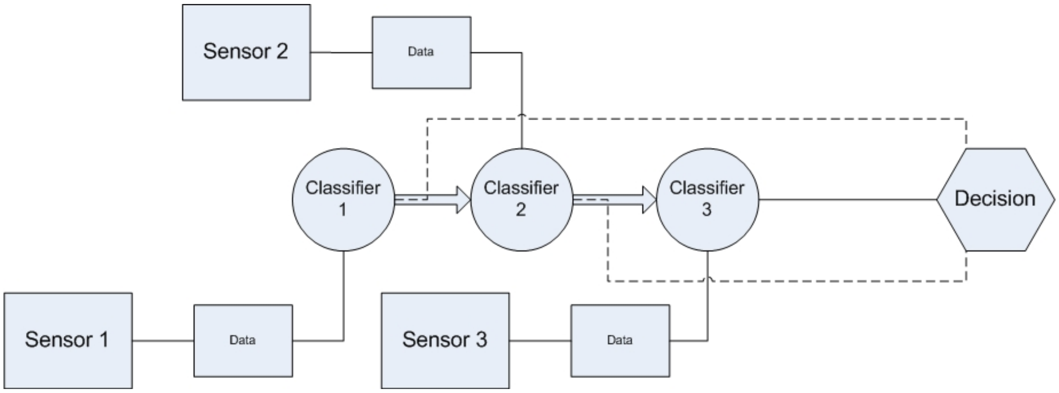


Figure 32. Illustration of a cascaded classifier.

- <u>Voting Fusion</u> – There are several different approaches to high-level fusion via voting. First is the simple voting scheme, in which we take the decision made by the majority of the feature vector classifier combinations as the decision. Similarly, we can set up a weighted voting scheme, in which our a priori probability of correct classification of each classifier and feature vector combination is taken into account. The general high-level fusion method illustrated in Figure 33 below is an example of a simple voting fusion method.

Advantage: This method can employ each sensor to it best use because each classifier can be tuned to the particular data from each sensor. Further, any domain knowledge that we may have as to the effectiveness of a sensor given the environmental conditions can be used to weigh each sensors vote.

Disadvantages: Deciding which sensors are best for what use and which feature vector combination is best can be a bit of a problem. This requires a significant amount of prior knowledge of the output of the sensors and the characteristics of the target.

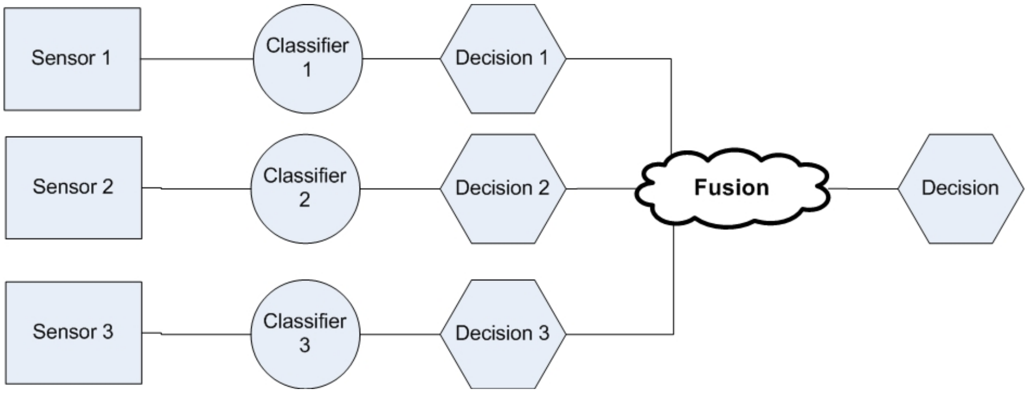


Figure 33. Illustration of high-level fusion via voting.

There are several advantages to the use of high-level sensor fusion. First, because the feature vector and classifier combination is mostly independent, this approach to sensor fusion can be highly parallelizable. We can run each classifier on a separate machine or as separate, concurrent thread to take advantage of the current tread towards parallel architecture. This method best captures data and decisions between passive and active sensor data. It does not make sense to fuse LIDAR data and thermal IR in a low-level sense, without some prior processing that results in an approach that is just below high level fusion. High-level fusion also has the distinct advantage of not requiring the same level of registration of sensor data that is required in low-level approaches. Temporal registration, depending on the context, may also be less important than in low-level approaches.

Utilizing sensors according to their best uses is a large part of the hypothesis of this thesis. In keeping with that concept, the first part of this experiment is designed to measure what benefit may be gained in terms of efficiency of detection from color alone to color focused by IR.

2. Our High-Level Approach

In our high-level approach, we use the IR imagery as an object location hypothesis generator. Our IR imagery is returned from the sensor as a single channel grayscale image. This imagery is, basically, an intensity image where the intensity represented in each pixel is the relative heat of the object, relative to the temperature of the calibration source, at that place in the sensor's field of view. Assuming our objects of interest radiate more heat than the environment, it is simple to threshold the single channel image to create likely locations of objects. We can then use this threshold image to create a binary image of object/no object locations. It is our hypothesis that the location of objects in this image can be used, because of our spatial and temporal registration, in our color imagery to focus our search for objects of interest. This can greatly limit the number of pixels that must be searched.

3. Methodology

After capturing our color and IR imagery, we deinterlace our imagery as described in chapter IV. We then register our infrared imagery to our color imagery using common point correspondences to create a homography matrix allowing us to project the IR image coordinates into the color coordinate system. With this registered data, we process our IR imagery by creating a series of thresholds, which represent likely levels of radiated heat by vehicles in our field of view. The poor quality of IR image as shown in Figure 34, results in a small range of thresholds which produce desired results. Example images from the thresholds explored are shown in Figure 35 through Figure 38 below. The tilted line on the right edge of all the IR images is an artifact of the spatial registration process, described in Chapter III. It shows that, despite best efforts to the contrary, the image fields of view and orientations are not exactly the same in the raw videos. The color image that corresponds to the IR image from which these threshold images are taken is shown in Figure 39.

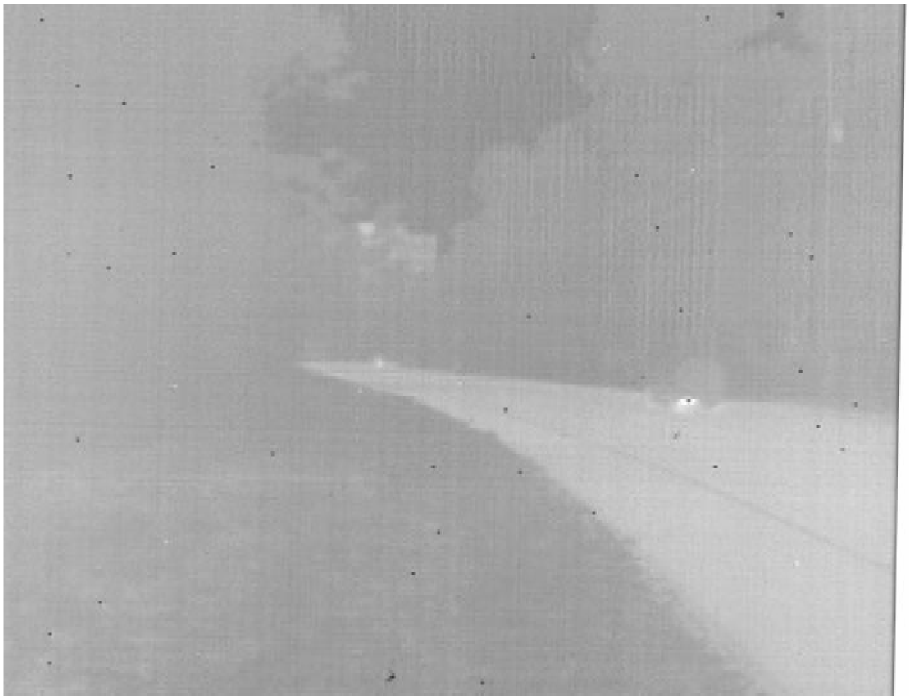


Figure 34. Original registered IR image.

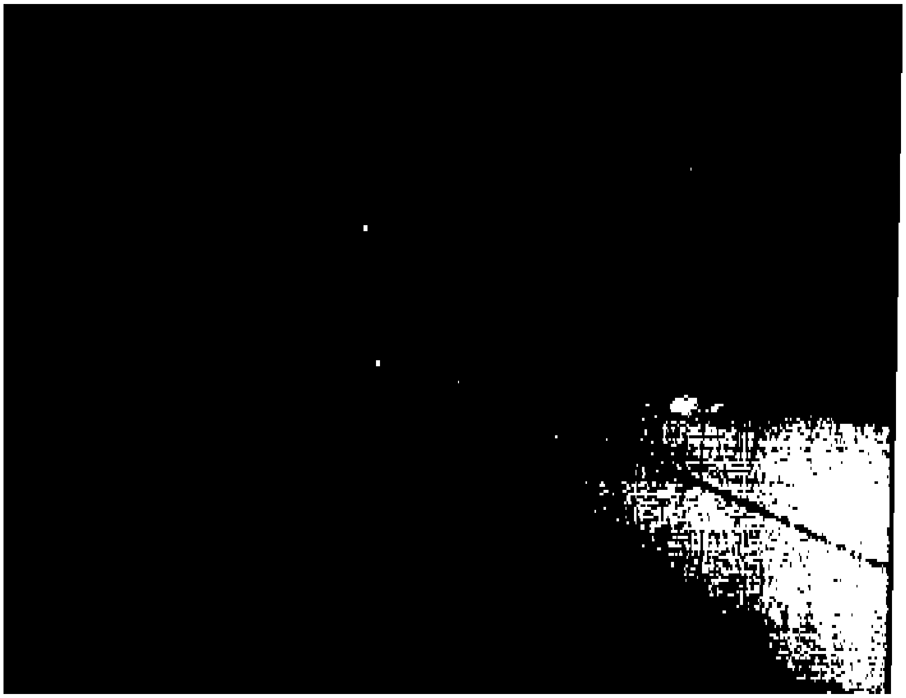


Figure 35. Threshold image with a threshold value of 200.

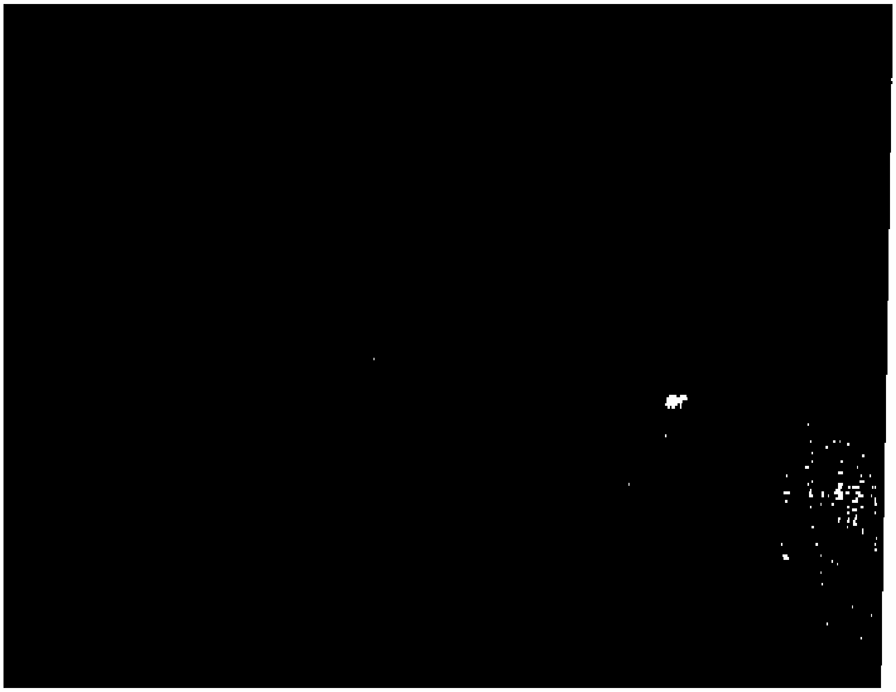


Figure 36. Threshold image with a threshold value of 210.

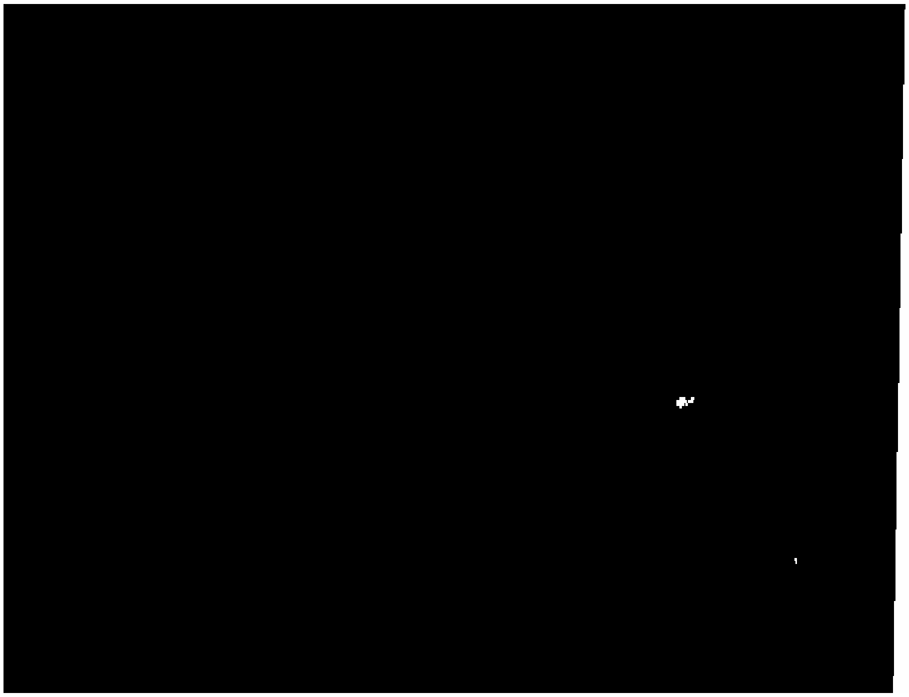


Figure 37. Threshold image with a threshold value of 220.

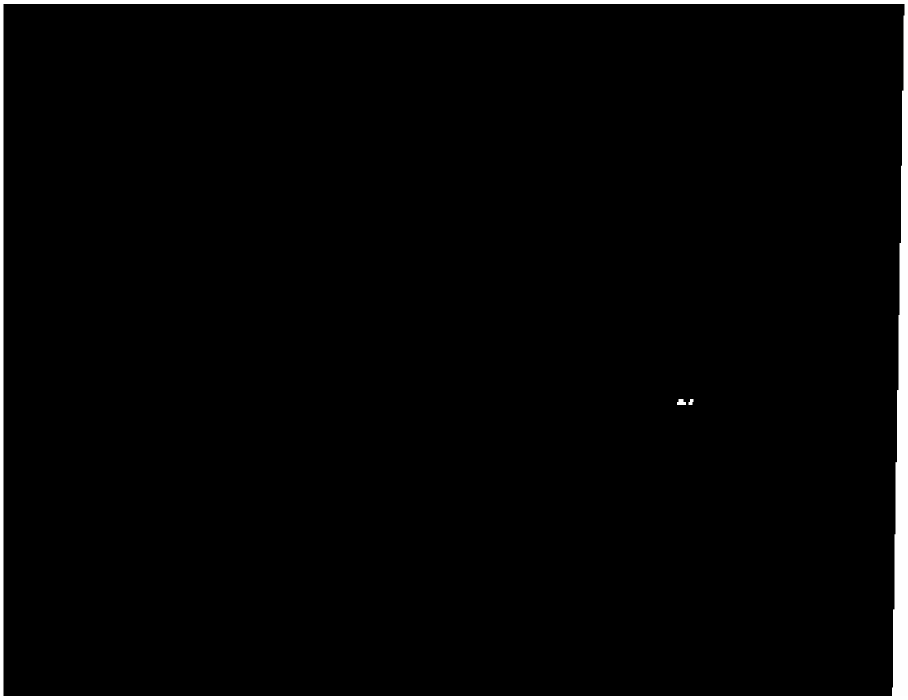


Figure 38. Threshold image with a threshold value of 230.

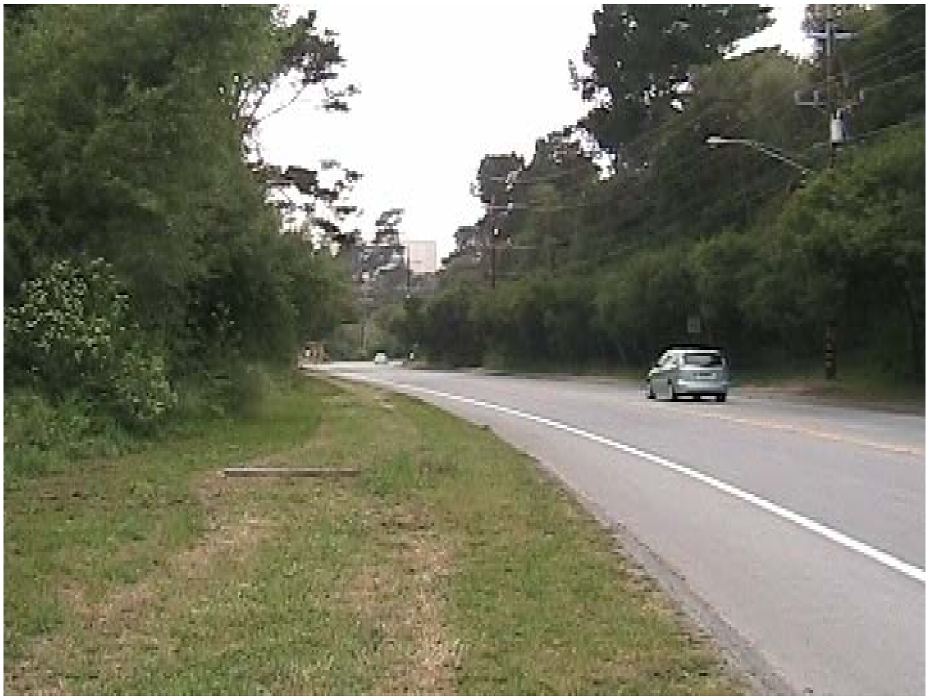


Figure 39. Original color image.

We apply morphological operations to the threshold imagery to reduce the effects of noise and to decrease the number of small pixel size hotspots that are the result of relatively warm, but not hot, areas of the scene close to the camera, which can be observed in the lower right corner of Figure 38. The resulting image is then made binary, in which any pixel with a value of greater than 0 (not black) is hypothesized as an object location. Figure 40 is an example of the results of these morphological operations.

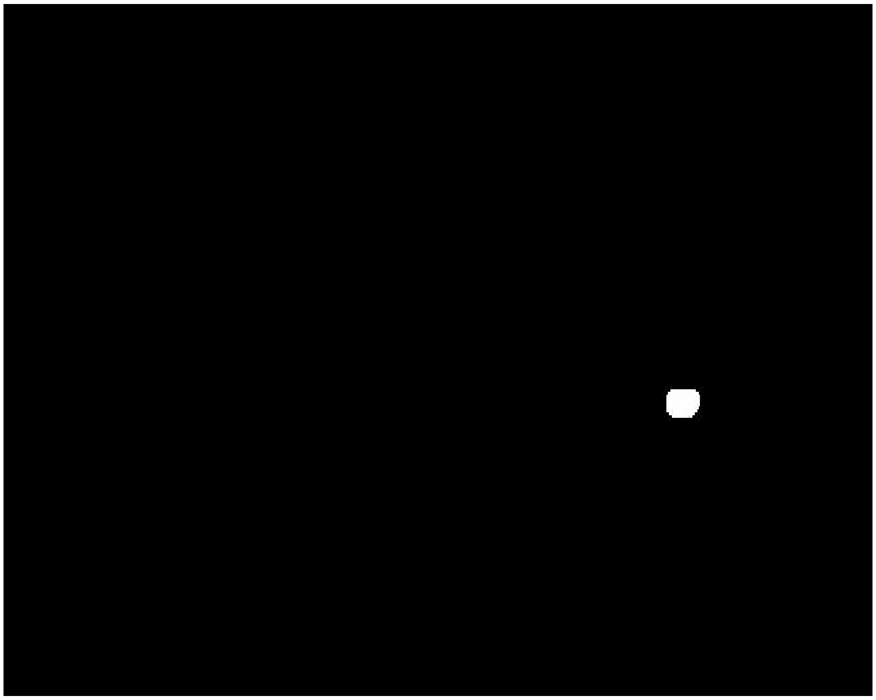


Figure 40. Binary results of morphological operations.

Clearly, the image in Figure 40 reflects the location of the vehicle in the color image, Figure 39, which was taken at the same time by the color camera. Other images are similar in locations predicted. However, smaller vehicles and vehicles that were not sufficiently hot did not show up well.

Once we have a prediction image, such as the one shown in Figure 38, we use the connected components, the white areas, to extract an ROI in the color image to check our predictions. In the beginning, our prediction areas were limited to areas that were hottest on the vehicles. Often times these locations did not correspond to the whole vehicle. Examples of these areas are shown in Figure 41.

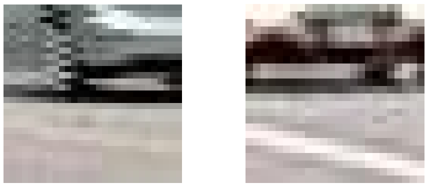


Figure 41. Predicted locations of vehicle before domain knowledge.

By expanding the area around the hot spots we were able to get more consistent results in the color imagery from the predictions. Here we utilized the domain knowledge that generally the hottest part of a vehicle is the area under it, including the tires. Examples from the predictions after application of this knowledge are shown in Figure 42 and 43. Figure 42 is the capture from the IR prediction shown in figure 40. Note that the imagery used for this hypothesis generation is pre-deinterlacing. All other imagery in this experiment is deinterlaced

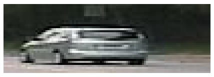


Figure 42. Color image location of hypothesis shown in IR image above.

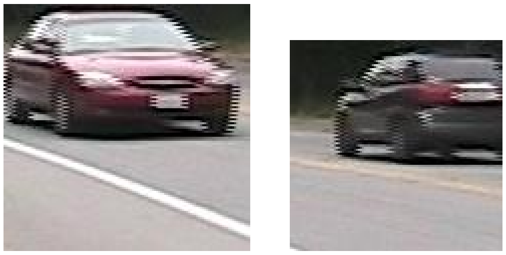


Figure 43. Examples of predicted location of a car and an SUV.

We must verify the locations hypothesized above. There are a variety of means to verify locations. In our case, with a small number of training images, we verified our predicted locations by hand.

4. Results of High-level Detection

The images shown in the section above are indicative of the best results. Given the poor quality of our IR images, we received encouraging results. Our hypothesis generation module using IR alone consistently generated correct hypothesis locations for vehicles over approximately 40 x 40 pixels. Vehicles in the field of view that are smaller than 40 x 40 pixels are often missed in the generation of hypotheses. This inability to detect vehicles smaller than this size is primarily due, we believe, to two factors. First, our IR camera is calibrated by one point, cold, or two point, hot and cold, references. The morning we captured our images we calibrated the camera using only a one-point reference. Second, the proximity to the point of thermal crossover, resulted in imagery which appeared to have very little in terms of heat differentiation, even in terms of vehicles which were warm in comparison to the environment.

Another issue that came up in this approach is the variety of areas that show heat on and around a vehicle. There are times in which the roadbed reflects heat from the underside of a vehicle and other times when the hottest part of a vehicle is the tires, for instance. Given the various areas, it is difficult to create a bounding box that is tight to the vehicle in every situation. The results of this variation is shown in the different sized boxes in Figure 42 and 43.

Even with these restrictions and issues, we detected about 80 percent of vehicles greater than our size restriction of 40 x 40 pixels. Lesser than our size restriction, we only detected 40 percent of our vehicles. We suspect that with a better camera and better calibration procedures, detection using this method should improve considerably. This would result in needing to search much less of the whole image in color than would be necessary without it. Detection using the whole image is the topic of the second part of experiment two, in the next section of this chapter.

D. DETECTION USING LOW-LEVEL FUSION

1. Introduction

Low-level fusion is the default approach for object detection used by many computer and electrical engineers. Low-level fusion is illustrated in Figure 44. For the digital signal processing practitioner, it may be the fusion of actual, continuous, signals received from several antennae. In the case of images, low-level fusion takes the form of some combination of raw pixel values of both images.

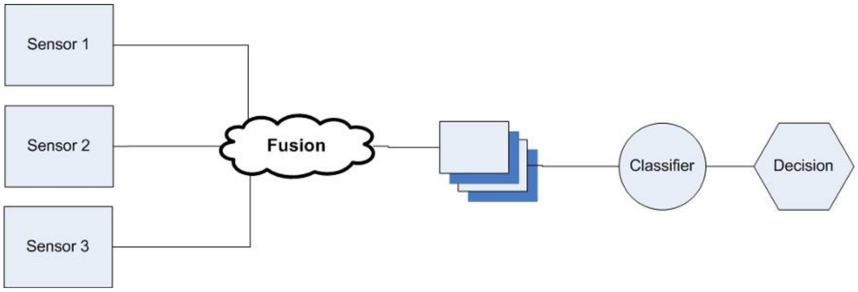


Figure 44. Illustration of low-level fusion.

Example approaches to low-level fusion of images include(Hall and McMullen 2004):

- <u>Averaging across channels</u> – Assuming each image has the same number of channels, each channel from each image can be added together and simply divided by the number of images received by the sensor suite. Averaging across channels is illustrated in Figure 45, below.

Advantage: The results can be viewed directly as a regular image.

Disadvantages: Information can be lost due to the transformation from n dimensions to n-k dimensions. Similarly, an image resulting from this fusion approach may not make sense when viewed.

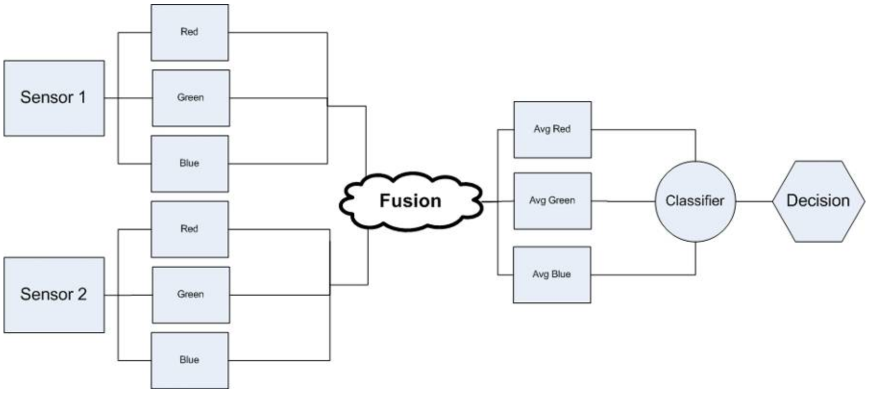


Figure 45. Illustration of averaging over all channels.

- <u>Replacement of a channel with like information</u> – An example of this type of fusion is the conversion of an RGB color image into the HSV color space, in which the V channel roughly corresponds to the intensity, or lighting conditions, of the scene. Then we can replace the V channel by a single channel image, such as an image from a thermal IR sensor. Replacement of a channel with like information is illustrated in Figure 46 below.

Advantages: First, this method can be the least computationally intense. This method also returns an image that is directly viewable, with the results being perhaps more sensible than the previous method.

Disadvantages: We do not know what information is most important to any classification of our data. It is likely that we could throw away important data by using this and the previous technique.

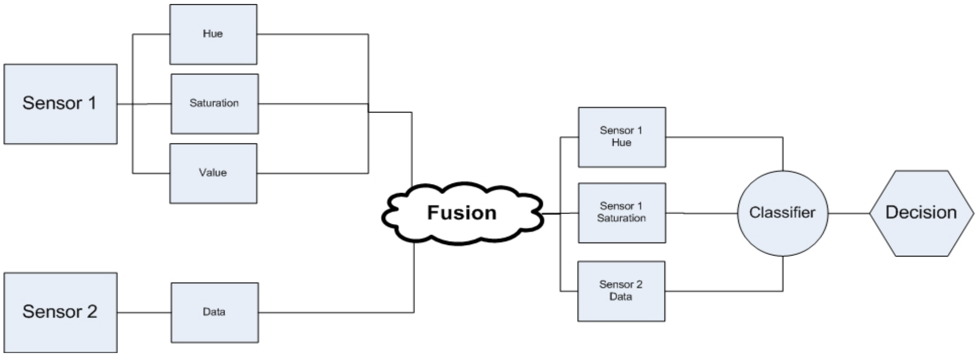


Figure 46. Illustration of channel replacement.

- Hypercube – Typical of many approaches in remote sensing and geographic information systems, this method simply adds all channels of all sensor images upon each other. If we have a one channel thermal IR, n x m image and which to fuse it with an RGB color, n x m, image, we put them together into an n x m x 4 channel image. Fusion via hypercube is illustrated in Figure 47 below.

Advantages: No data is lost; the most important data is preserved. With all the data, it is easier to develop unique relationships between the data, so that we may be able to apply data reduction techniques, such as principle component analysis, which will be discussed in a later section.

Disadvantages: Due to the larger amount of data, this technique is the most data intense of the low-level fusion techniques. Once we apply more than three channels, our image is no longer directly viewable on standard display hardware.

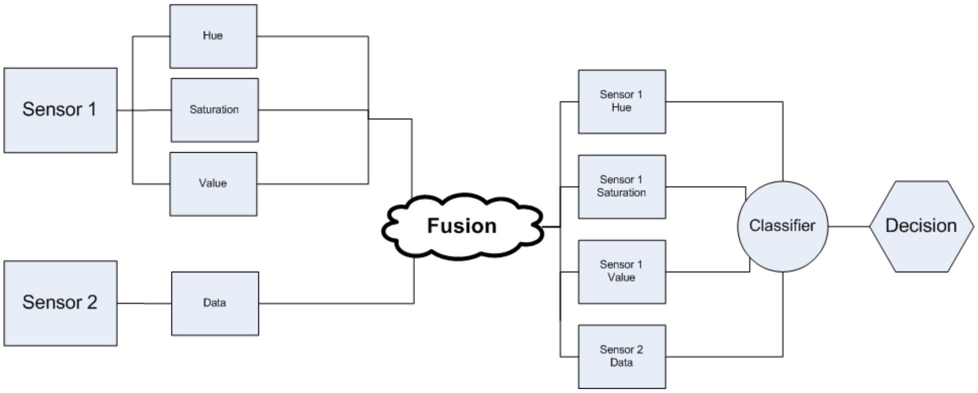


Figure 47. Illustration of fusion via hypercube.

Besides the specific benefits listed in each approach to low-level fusion above, there are several general advantages of low-level fusion. First, these approaches only require one feature vector to represent all images in the sensor suite. Another major advantage of low-level fusion is that we only require one classifier to classify each frame of our input. There are approaches in low-level fusion that use more than one classifier, but these tend toward the gray area between low-level and high-level fusion. Because only one feature vector and one classifier are required, low-level fusion approaches can be less computationally expensive than later methods.

2. Our Low-Level Approach

Our low-level approach to detection uses a scanning approach to generate object location hypotheses. The hypothesized location is simply a ROI of the image that is used to measure the variance normalized distance from a known set of vehicles in eigenspace. The ROI is scanned across the image row-by-row and column-by-column. In this part of our experiment, we perform this hypothesis generation on our two raw image sets, color and IR, and 11 different fusion types. The eleven fusion types, with corresponding example images, except for the hypercube which cannot be displayed by conventional means, are:

- R, G and B channel replacement with IR, shown in Figure 48.
- R, G and B channel averaging with IR, shown in Figure 49.

- All color channels averaging with IR, shown in Figure 50.
- V channel averaging with IR, shown in Figure 51.
- V channel replacement with conversion back to RGB color space, shown in Figure 52.
- V channel replacement with conversion back to RGB color space, shown in Figure 53.
- Hypercube of R G B of color and grayscale IR

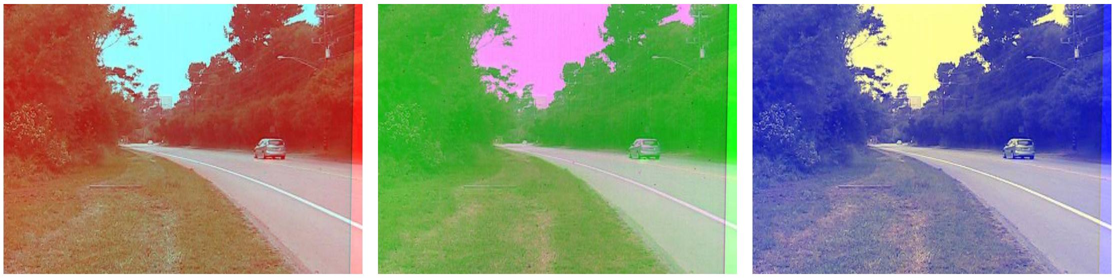


Figure 48. R, G and B channel replacement with IR in color.

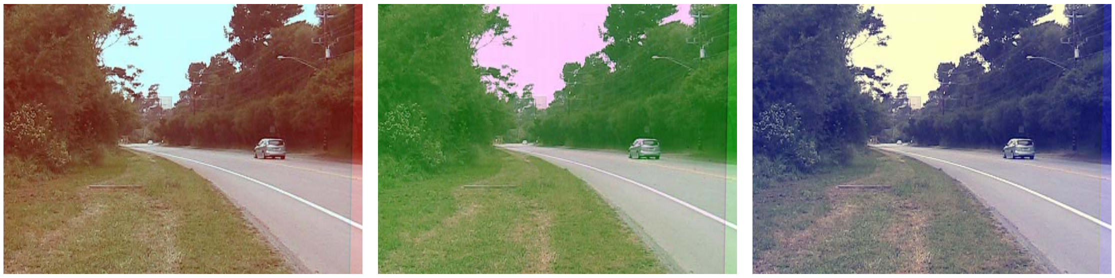


Figure 49. R, G and B channel averaging with IR.

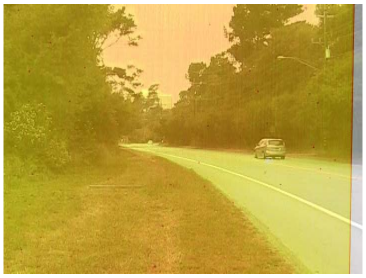


Figure 50. All color channels averaging with IR.

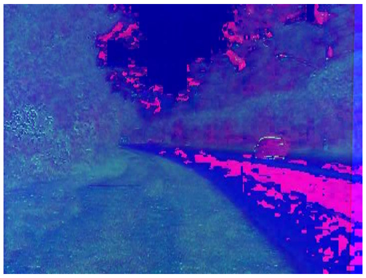


Figure 51. V channel replacing with IR in HSV color space.

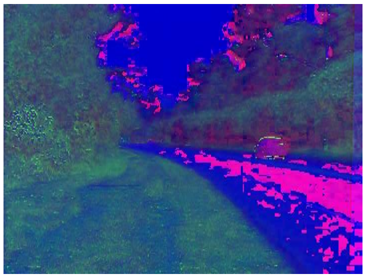


Figure 52. V channel averaging with IR in HSV color space.

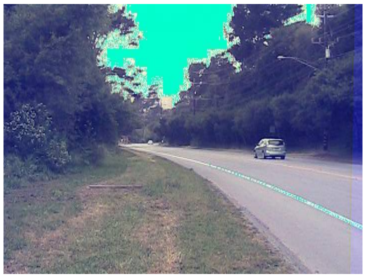


Figure 53. V channel replacement in HSV color space with conversion back to RGB color space.

3. Methodology

We capture and process our video as we did in the previous experiments. We then deinterlace the frames as before. We then hand annotate our training imagery with locations of vehicles. We extract the sub-image of the annotated vehicle. We have 55 hand annotated vehicles in the training set. We also select 20 random frames from our video, that do not contain vehicles, which will be used to capture known negative examples in later processing. These frames are deinterlaced and processed in precisely the same way as the previous imagery. We use these frames to create 200 negative examples as described below.

We grab a random appropriately-sized sub-image from our set of negative frames. Our positive training images have a 1.500 +/- .05 aspect ratio. Therefore, we process our negative training examples so that they all have an aspect ratio of 1.500. We restrict our negative sub-images to range in size from 20 pixels wide to 150 pixels wide. This range relates to the smallest vehicle we have been able to recognize and detect to the largest vehicle in any frame in our video. The last step in creating our negative training set, is to resize the negative examples to 16 by 48 pixels. We perform this step by Gaussian smoothing and sub-sampling as described in Chapter III. These are our negative training examples.

We next create the eigenvector and eigenvalue set, using PCA as explained in Chapter III. This set of eigenvectors is created using only positive training examples.

To project an image into eigenspace, we first subtract out the mean image of all the positive training ROIs from the dataset associated with that eigenspace. Remember that the mean image represents the origin of our eigenspace.

$$ROIMinusMean = \begin{pmatrix} negImg_{pixel1} \\ negImg_{pixel2} \\ \vdots \\ negImg_{pixeln} \end{pmatrix} - \begin{pmatrix} \mu_{pixel1} \\ \mu_{pixel2} \\ \vdots \\ \mu_{pixe\ln} \end{pmatrix}$$

Then multiply the transpose of the associated set of eigenvectors, where **m** is the number of eigenvectors and **n** is the number of pixels represented in the image vector, by this result

$$projected = \begin{pmatrix} evec_{11} & \dots & evec_{1n} \\ \vdots & \ddots & \vdots \\ evec_{m1} & \cdots & evec_{mn} \end{pmatrix}^T * ROIMinusMean = \begin{pmatrix} distEVec_1 \\ distEVec_2 \\ \vdots \\ distEVec_m \end{pmatrix}$$

The values of the projected image are simply the distance along each associated eigenvector from the origin of eigenspace to the location of the projected image in eigenspace.

We measure the distance from our negative hypothesis test locations to the origin of this set of eigenvectors using the Mahalanobis distance. Our positive training set is our known set. Our negative training set is an unknown set in terms of variance. We also use the Mahalanobis distance because it is a variance normalized distance measure. PCA creates the eigenvector basis based on the directions of most variation in the data set, therefore a variance normalized distance measure appears to be the best fit. We tried several different distance measures which all resulted in poorer results.

By dividing the distances calculated in the last equation, element-wise, by the associated eigenvalues, we get a variance normalized distance form the origin of eigenspace.

$$\begin{pmatrix} MalDistEVec_1 \\ MalDistEVec_2 \\ \vdots \\ MalDistEVec_m \end{pmatrix} = \begin{pmatrix} distEVec_1 \\ distEVec_2 \\ \vdots \\ distEVec_m \end{pmatrix} ./ \begin{pmatrix} EVal_1 \\ EVal_2 \\ \vdots \\ EVal_m \end{pmatrix}$$

Thus, the Mahalanobis distance mitigates the effects of any great distance in any sub-set of the eigenvectors by the variance associated with that distance. The result is that the distances that result from the eigenvectors that encapsulate the most variance are amplified and those that result from the eigenvectors with the least variance are subdued.

Once we have our distances to each negative example, we measure the distances to the positive examples. To measure these distances, we create a leave-one-out eigenvector set, using all raw positive examples except the one which we leave out. If we have a large enough training set, then the effects of leaving one out should be minimized. We use the same methods described in the negative examples section above to project the image into eigenspace and measure the Mahalanobis distance from the eigenvector set to the positive image. With these distances, we create histograms of negative distances and positive distances to visualize the distance measurements of positive and negative examples. All histograms are included in Appendix A. The most descriptive are shown in the results section that follows.

With our negative and positive distances, we can perform a comparison using a threshold distance to determine the best fusion method. We vary the threshold from the minimum of all distances associated with that set of eigenvectors and fusion type to the maximum of all distances associated with that same set and fusion type. We vary the threshold by adding 1/600th the distance between the minimum and the maximum. This gives us 600 measurement points with which to compare.

We use this varying threshold as a linear boundary below which we classify our sub-images as a vehicle detected, above which we classify our sub-images as no vehicle detected. A known positive sub-image that is classified as a detection is added to the true positive count. A known negative sub-image that is classified as a detection is added to the false positive count. These counts are then divided by the total positive and total negative, respectively, to give us the true positive rate and false positive rate.

We compile and plot these error rates to create ROC curves to analyze the performance of the thresholds and fusion types in the next section.

4. Results of Low Level Fusion

Initial results from low-level fusion at first seemed incorrect. In PCA, on an ideal, symmetric data set the area under the ROC curve should increase as we increase the number of eigenvectors used to create that ROC curve. In this case, by symmetric we mean there is an approximately equal number of images at approximately equal distances from the origin of eigenspace along each eigenvector. With a data set that is symmetric in this way, we would expect to have a symmetric cloud of positive examples centered at the origin of eigenspace, which is a sub set of the larger symmetric cloud of all examples including those negative examples. If we have a data set that is symmetric in this way, we expect to see the greatest area under our various ROC curves to be associated with the greatest number of eigenvectors. That is not necessarily the case in our dataset. The issue, we hypothesize, is the fact that our data set is probably not symmetric.

Symmetricity is not easy to measure in the high dimensionality space that we are operating in. However, it is easy to see how our data set could be less than perfectly symmetric. We have an unbalanced number of cars versus trucks. We have an unbalanced number of vehicles that are approaching versus those that are departing from the sensor locations. Equally important, as we project them into our eigenspace, and probably even less symmetric is our great number of negative examples, from random portions of the scene.

Use of IR in replacement or average of channel with highest overall impact resulted in good results. In our case, the channel with the highest values was the green channel, because of the mostly green background. By replacing or averaging this channel with IR, and using PCA with the Malhalanobis distance, significant gains were made in terms of detection error rates over that of color alone, or any other fusion method. The ROC curve for replacing the green channel using 53 eigenvectors is shown in Figure 54 below. The histogram of positive and negative distances is shown in Figure 55. In all histograms, the blue columns represent distances associated with positive examples and the red column represent distances associated with negative examples. An ideal histogram would have all blue columns as a distinct set easily distinguishable, with some distance, from all red columns.

The ROC curves for IR and color are shown in Figures 58 and 60. The histograms for IR and color are shown in Figures 59 and 61.

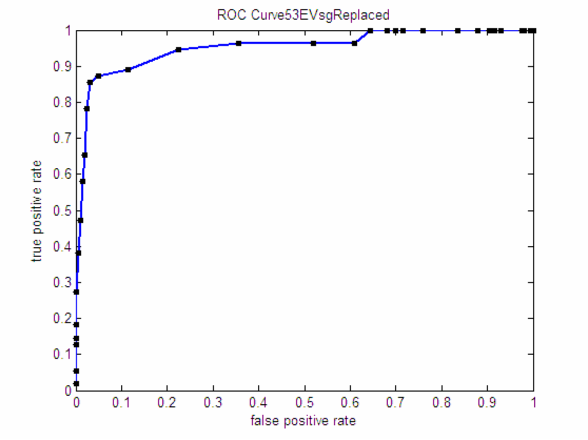


Figure 54. ROC curve for G channel replacement using 53 eigenvectors.

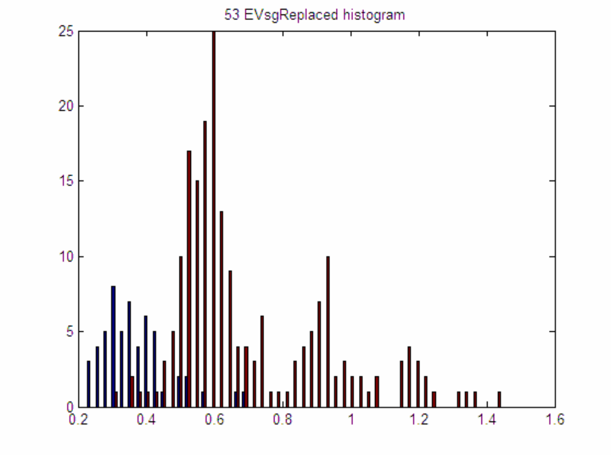


Figure 55. Histogram for G channel replacement using 53 eigenvectors.

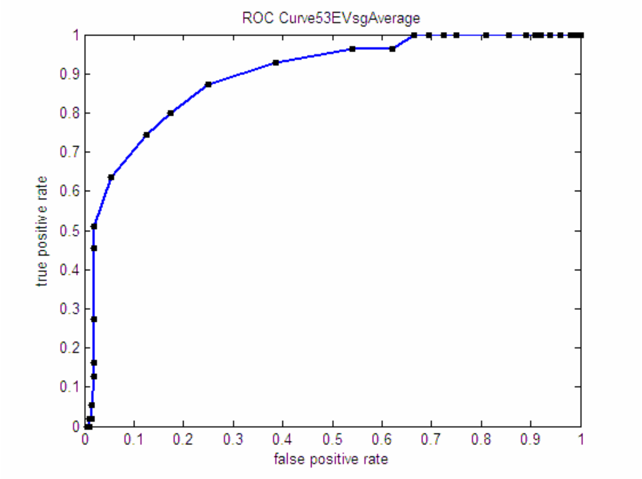


Figure 56. ROC curve for G channel averaging using 53 eigenvectors.

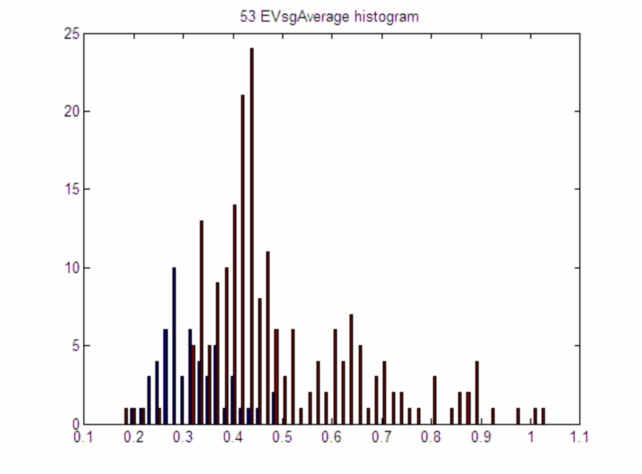


Figure 57. Histogram of G channel averaging with 53 eigenvectors.

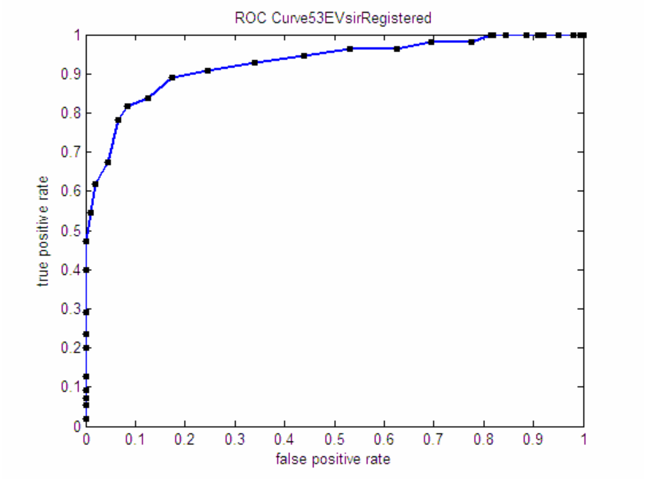


Figure 58. ROC curve for IR based on 53 eigenvectors.

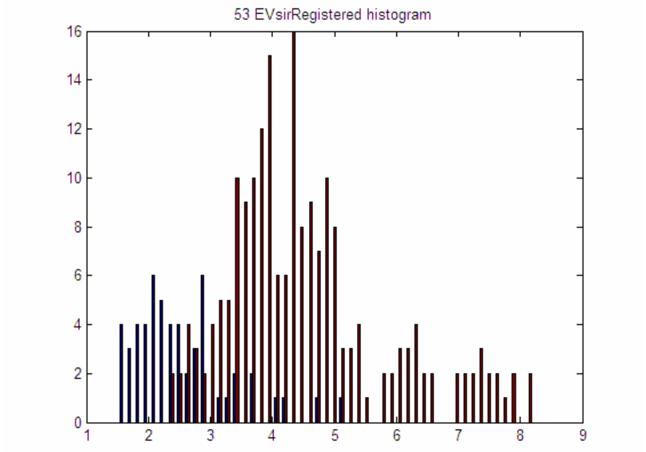


Figure 59. Histogram of positive and negative distances from 53 eigenvectors in IR.

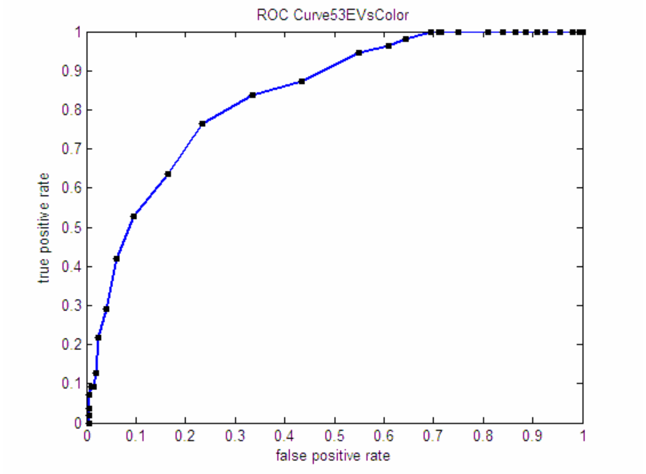


Figure 60. ROC curve for color detection using 53 eigenvectors.

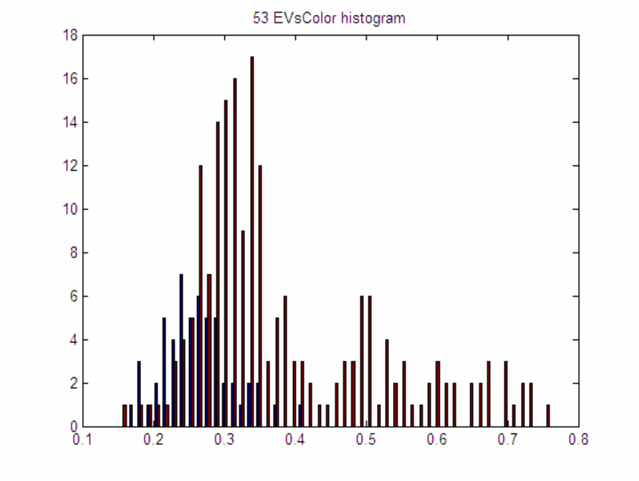


Figure 61. Histogram of positive and negative distances from 53 eigenvectors in color.

5. Discussion

We do not know if these results would generalize to different scenes. For instance, if our background is a cityscape that is mostly gray, then would replacing a single channel do as well as averaging all channels or as well as color alone? If our background was ocean, would replacing the blue channel do as well as averaging over all channels in color? We hypothesize that averaging over all channels would perform the best given any general situation by tempering the effects of any spike in color due to a background that is mostly a single color. The replacement of a channel may produce better results, but replacing a single channel may not be a general enough solution as any change in background would probably result in loss of benefit from the channel replacement. A solution may be to perform the channel replacement in real time as the background changes. The ROC curve for G channel averaging with IR using 53 eigenvectors is shown in Figure 56 above. The histogram for G channel averaging is shown in Figure 57. It is readily apparent that this averaging of a channel is not as good as the channel replacement method

The channel replacement method was explored here to check the impact on the detection and recognition of vehicles of an individual channel. What we did not know, was whether one channel contributed more in terms of noise than any other. That may be why we see a better result from the replacement of the G channel above. If the dominant channel is the G channel, then the G channel may be responsible for more than one third of the noise in the three-channel image. If this is the case, then replacing that channel may filter out that noise while also not significantly decreasing the useful information because of the impact of the IR channel replacement.

The second best result came from our registered IR imagery. This result was unexpected until further analysis of the IR imagery. Perhaps because of the poor quality and resulting lack of background detail, see Figure 62 below, the vehicles stood out enough that the detection was somewhat trivial. Most areas in the IR imagery are very gray. Very few areas show the heat that most vehicles in the video show. This is a source of concern, because the results in this case may not be reproducible with the same error rates during a different time of day.

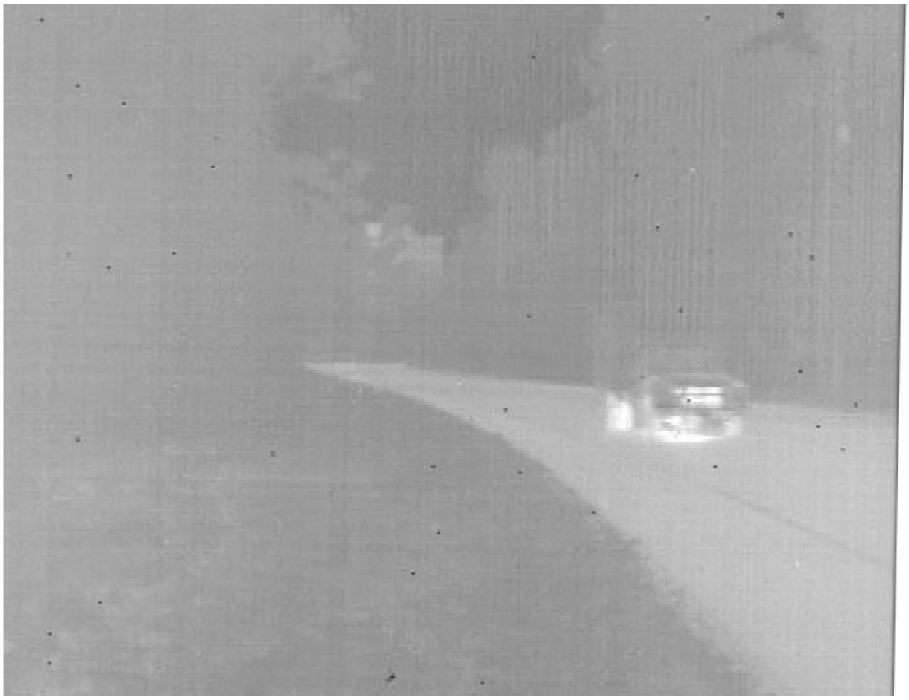


Figure 62. Typical IR image showing lack of background detail.

In Figures 41 though 43 in the results section above, we show typical results from thresholding that we performed in our high-level detection experiment. The threshold image shown below is not necessarily typical. Because we were so close to the point of thermal crossover, vehicles showed up very well. Figures 41 is based upon the base image shown in Figure 39.

Figure 63. Threshold image showing difference of vehicle and background.

Based upon the images and reasoning above, IR registered to color, performed very well on detection of vehicles. The ROC curve, shown in Figure 59 below, for 53 eigenvectors as well as the histograms of distances, shown in Figure 60, clearly shows that the differences in distances between negative and positive examples is better than any other method of either raw or fused detection. Remember that positive distances are shown in blue and negative distances are shown in red.

Raw color performed reasonably well, but not nearly as well as either G channel replacement or raw IR. We believe the primary reason for the difference in detection rates is due to the large amount of green foliage in the scene. The ROC curve and histogram are shown in figures 60 and 61 above for comparison purposes.

E. SUMMARY

This chapter has shown that using high-level fusion, infrared sensors can limit the area that must be searched in color imagery. This focusing can also result in better error rates in color by culling many of the areas that may create false positives. Detection using low-level fusion can be made more accurate by using IR either alone, or by using IR as a replacement for the channel in RGB that contributes the most. Presumably this is due to the increase in noise from the largest contributor to the image.

Given our data above, there are clear advantages to the use of high-level fusion. These benefits include more rapid detection, by limiting object hypothesis areas, and fewer false positive detections by limiting the number of validation areas, as shown in Figures 41- 43 above, to those that are detected by our IR sensor. Most areas that are hypothesis locations in color, based on IR hypotheses, are correct locations. Therefore, hypothesis validation of troublesome areas that may have been triggered by use of color alone is negated. This method could be utilized in a cascaded classifier in which we use IR first and then use color only when we have utilized IR to develop complete hypothesis locations.

In the next chapter, we discuss some of the conclusions we have reached in this thesis, as well as a few recommendations for further work. Appendix A contains all of the ROC curves for all fusion types using 53, 50, 40, 30 and 20 eigenvectors for comparison purposes. Appendix B contains all of the histograms for positive and negative distances using the same numbers of eigenvectors.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND SUMMARY

A. OVERALL RESULTS

This study served as an introduction to sensor fusion using eigenspace-based techniques. Further, we showed that using sensors other than color can result in gains in terms of error rates in detection of vehicles, as shown in the comparison of the ROC curve for detection using IR shown in Figure 58 and the ROC curve for detection using only color, shown in Figure 60. There is also a significant gain that can be made in terms of search speed when using IR to focus color detection in a high-level manner.

It is clear from this work that recognition using color alone works better than other sensors given scenes such as ours. What is not clear is if, in a more cluttered environment, the addition of another sensor would help in the recognition of vehicles. When we add thermal IR to color, our recognition rates go down, but not by much. If we have an environment in which IR works better than color, such as at night or in low light conditions, the inclusion of thermal IR could increase the recognition rate over that of color alone. Results from our detection experiment show that in an environment with the right conditions, IR helps considerably in the detection of vehicles. This should apply to recognition of vehicles as well.

B. DISCUSSION

Our results in recognition using low-level sensor fusion show promise. The low error rates in recognition using fusion of IR and color demonstrate the usefulness of low-level sensor fusion. If we have a situation in which IR is optimized, the fusion of IR and color would probably result in better recognition than color alone. Color imagery suffers from many environmental effects that thermal IR is resistant to, such as smoke, fog and lack of illumination. Thermal IR is not immune to these environmental effects. However, the effects of these environmental aspects are much less pronounced in IR than color. There are also a large number of sensors on the market that exhibit even better results against many of these environmental effects than IR. Many of these sensors may give even more benefit than using IR and color alone.

The results of our detection experiment are noteworthy in that the replacement or averaging of the channel with the greatest overall impact on the color image showed itself to be the best fusion method. Granted, the general utility of this method requires more exploration, as noted in Chapter IV and in the future work section below. However, utilizing this method in a general computer vision system in which frame-by-frame decisions are made concerning which color channel to replace is feasible and may result in general benefit.

This method is a low-level method, but can be used in conjunction with a high-level sensor fusion such as that shown in Chapter IV, where IR is used to focus search areas for detection of objects in color.

There are numerous scenarios in which one particular sensor is optimal for a certain scene. Given a large number of sensors, it is possible to optimize detection and recognition of objects of interest based upon that sensor input. However, conditions can change in a scene from moment to moment such that one sensor may be optimal one moment, but not at all the next. Sensor fusion could provide the best results in these situations by providing input to recognition and detection schemes that allow for the use of the best sensor input. Experiment two bears this out in the increased detection rate of vehicles by the replacement and averaging of the green channel using IR. A general scheme to replace a particular channel or sensor's input with another may not be possible though. It appears that the increase in detection rates with this type of fusion is the result of the mitigation of the green background and it's effect on our classification methods. It is unlikely that the replacement of the predominant channel would result in better detection given any general scene. It is easy to imagine a scene in which the predominant color is the primary discriminator of an image. In this case, the fusion describe above would result in worse detection rates, rather than better.

It is important to note that all methods of detection and recognition are in some measure on the type of object with which are most relevant to the task at hand. Still, most objects exhibit some difference in temperature with their background. This difference in temperature can be exploited in the same manner as it is in our high-level IR focus method.

C. FUTURE WORK

This thesis demonstrated that gains can be made through the use of fused sensor data. Several recommendations for further work include the following:

1. Purchase Collocated, Spatially Synchronized Thermal IR and Color Cameras

Current synchronized cameras that include thermal IR and color far exceed that of those used for this study in terms of spatial synchronization. Though the synchronization routines that we used in this study were acceptable, given the conditions of speed and size of vehicles, it is likely that further use of these techniques would require greater synchronization in terms of time and space. It is likely that objects of interest in further study would not be limited to below 45 miles per hour. It is also unlikely that further objects of interest would likely be always greater than the minimum size we established in this work for detection, regardless of method used. In order to ensure the best temporal synchronization, at least one of the cameras needs a synchronization, genlock, input in order to temporally sync it with the other.

2. Implement Sensor Fusion with Viola-Jones Detection

Viola-Jones detection (Viola and Jones 2001) was one technique discussed as a possible approach to sensor fusion, before we settled on using simple PCA. Given the power of Viola-Jones, it is likely that it could result in better detection and recognition, especially in high level fusion, while minimizing the effects of greater amounts of data from the use of multiple sensors. If raw IR were used as a beginning stage, much of the complexity of searching and detection in color may be mitigated.

3. Utilize a Non-linear Classifier to Obtain a Close to Optimal Classification

The classification in the detection portion of this thesis used a simple variance normalized distance measure, a linear classifier. The classification from this method is likely not optimal, especially given the non-linear and non-symmetric nature of this data set. The use of a non-linear classifier on data sets such as this and on fusion results like those in this thesis, could result in better error rates. There will be an increase in complexity, but perhaps not a great one.

4. Explore the Generality of Channel Replacement or Averaging with Channel of Greatest Impact on the Scene

As described in this thesis, the best results in detection resulted from the replacement of the green channel with IR. This was likely due to the great number of green pixels in the color imagery. A logical next step would be to explore if similar results can be obtained from scenes in which a different color is predominant, or the case in which no color is greatly predominant.

APPENDIX A: RECOGNITION RATES GIVEN KNN AND VARYING NUMBERS OF EIGENVECTORS

The data in this appendix refers to the recognition of vehicles as shown and discussed in Chapter III.

	1EV	2EV	3EV	4EV	5EV	6EV	7EV	8EV	9EV	10EV	20EV	30EV	40EV	45EV
1 NN	0.7059	0.7451	0.9412	0.902	0.902	0.9412	0.9412	0.9412	0.9412	0.9412	0.9608	0.9608	0.9608	0.9804
2 NN	0.7451	0.7647	0.9412	0.9216	0.9608	0.902	0.9216	0.9216	0.8627	0.8824	0.8824	0.902	0.8824	0.8824
3 NN	0.7451	0.7647	0.9412	0.9216	0.9608	0.902	0.9216	0.9216	0.8627	0.8824	0.8824	0.902	0.8824	0.8824
4 NN	0.8039	0.8431	0.9412	0.902	0.9412	0.8627	0.8824	0.8824	0.8627	0.8627	0.8235	0.8627	0.8431	0.8431
5 NN	0.8039	0.8431	0.9412	0.902	0.9412	0.8627	0.8824	0.8824	0.8627	0.8627	0.8235	0.8627	0.8431	0.8431
6 NN	0.6863	0.8235	0.9216	0.902	0.9216	0.8627	0.8627	0.8627	0.8627	0.8627	0.8431	0.8824	0.8824	0.8824
7 NN	0.6863	0.8235	0.9216	0.902	0.9216	0.8627	0.8627	0.8627	0.8627	0.8627	0.8431	0.8824	0.8824	0.8824
8 NN	0.7255	0.8431	0.9216	0.902	0.9216	0.8627	0.8627	0.8431	0.8235	0.8431	0.8235	0.8235	0.8235	0.8235
9 NN	0.7255	0.8431	0.9216	0.902	0.9216	0.8627	0.8627	0.8431	0.8235	0.8431	0.8235	0.8235	0.8235	0.8235
10 NN	0.7255	0.8431	0.9216	0.902	0.902	0.8431	0.8235	0.8235	0.8039	0.8039	0.8235	0.8039	0.8039	0.8039
11 NN	0.7255	0.8431	0.9216	0.902	0.902	0.8431	0.8235	0.8235	0.8039	0.8039	0.8235	0.8039	0.8039	0.8039
12 NN	0.7647	0.7647	0.9216	0.902	0.8627	0.8431	0.8431	0.8235	0.8235	0.8235	0.8039	0.7647	0.7843	0.7843
13 NN	0.7647	0.7647	0.9216	0.902	0.8627	0.8431	0.8431	0.8235	0.8235	0.8235	0.8039	0.7647	0.7843	0.7843
14 NN	0.7843	0.8039	0.902	0.8824	0.8039	0.8039	0.7843	0.7647	0.7843	0.8039	0.7647	0.7451	0.7451	0.7451
15 NN	0.7843	0.8039	0.902	0.8824	0.8039	0.8039	0.7843	0.7647	0.7843	0.8039	0.7647	0.7451	0.7451	0.7451

Table 1. Color recognition rates given number of nearest neighbors and number of eigenvectors.

	1EV	2EV	3EV	4EV	5EV	6EV	7EV	8EV	9EV	10EV	20EV	30EV	40EV	45EV
1 NN	0.5882	0.6275	0.5686	0.7255	0.7255	0.7255	0.7451	0.7451	0.7843	0.8824	0.8824	0.902	0.902	0.902
2 NN	0.5294	0.4902	0.5686	0.6863	0.6863	0.7255	0.7843	0.8235	0.8039	0.8039	0.8431	0.8235	0.8235	0.8235
3 NN	0.5294	0.4902	0.5686	0.6863	0.6863	0.7255	0.7843	0.8235	0.8039	0.8039	0.8431	0.8235	0.8235	0.8235
4 NN	0.6667	0.6667	0.6078	0.7451	0.7059	0.7647	0.8235	0.8235	0.8235	0.8431	0.7843	0.8039	0.8039	0.8039
5 NN	0.6667	0.6667	0.6078	0.7451	0.7059	0.7647	0.8235	0.8235	0.8235	0.8431	0.7843	0.8039	0.8039	0.8039
6 NN	0.7059	0.6471	0.6471	0.7255	0.7059	0.7059	0.7451	0.7647	0.7647	0.7843	0.7059	0.7255	0.7255	0.7255
7 NN	0.7059	0.6471	0.6471	0.7255	0.7059	0.7059	0.7451	0.7647	0.7647	0.7843	0.7059	0.7255	0.7255	0.7255
8 NN	0.7059	0.6863	0.6863	0.7059	0.6471	0.6667	0.6667	0.6667	0.6667	0.7255	0.7059	0.7059	0.7059	0.7059
9 NN	0.7059	0.6863	0.6863	0.7059	0.6471	0.6667	0.6667	0.6667	0.6667	0.7255	0.7059	0.7059	0.7059	0.7059
10 NN	0.7059	0.7059	0.6863	0.6863	0.6863	0.6667	0.7059	0.7059	0.7059	0.7451	0.7255	0.7255	0.7255	0.7255
11 NN	0.7059	0.7059	0.6863	0.6863	0.6863	0.6667	0.7059	0.7059	0.7059	0.7451	0.7255	0.7255	0.7255	0.7255
12 NN	0.7059	0.7059	0.7059	0.7451	0.7059	0.6863	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059
13 NN	0.7059	0.7059	0.7059	0.7451	0.7059	0.6863	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059
14 NN	0.7059	0.7059	0.7059	0.7059	0.7059	0.6863	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059
15 NN	0.7059	0.7059	0.7059	0.7059	0.7059	0.6863	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059	0.7059

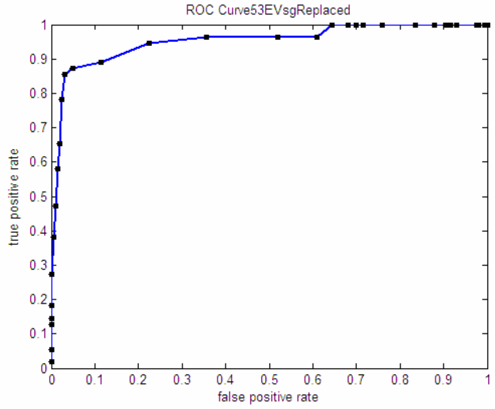
Table 2. IR recognition rates given number of nearest neighbors and number of eigenvectors.

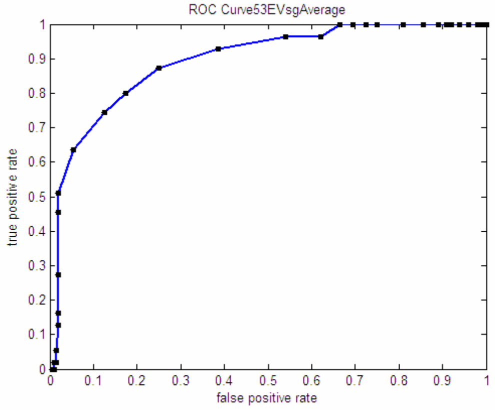
	1EV	2EV	3EV	4EV	5EV	6EV	7EV	8EV	9EV	10EV	20EV	30EV	40EV	45EV
1 NN	0.6863	0.8235	0.7843	0.9412	0.9412	0.8627	0.8235	0.8431	0.8627	0.8824	0.8824	0.8824	0.8824	0.8824
2 NN	0.8039	0.8431	0.7843	0.9216	0.902	0.8627	0.8431	0.8627	0.8627	0.8627	0.8431	0.8627	0.8627	0.8627
3 NN	0.8039	0.8431	0.7843	0.9216	0.902	0.8627	0.8431	0.8627	0.8627	0.8627	0.8431	0.8627	0.8627	0.8627
4 NN	0.8039	0.7451	0.8431	0.902	0.8627	0.8431	0.8627	0.8627	0.8431	0.8431	0.8039	0.8039	0.8039	0.8039
5 NN	0.8039	0.7451	0.8431	0.902	0.8627	0.8431	0.8627	0.8627	0.8431	0.8431	0.8039	0.8039	0.8039	0.8039
6 NN	0.7843	0.7255	0.7843	0.902	0.8431	0.8039	0.8039	0.8235	0.8235	0.8431	0.7843	0.8039	0.8235	0.8235
7 NN	0.7843	0.7255	0.7843	0.902	0.8431	0.8039	0.8039	0.8235	0.8235	0.8431	0.7843	0.8039	0.8235	0.8235
8 NN	0.7059	0.7451	0.8039	0.8431	0.8431	0.8235	0.8235	0.8039	0.8039	0.8039	0.7647	0.7843	0.7843	0.7843
9 NN	0.7059	0.7451	0.8039	0.8431	0.8431	0.8235	0.8235	0.8039	0.8039	0.8039	0.7647	0.7843	0.7843	0.7843
10 NN	0.7647	0.7059	0.8039	0.8431	0.8235	0.8039	0.8039	0.8039	0.8039	0.7843	0.7451	0.7451	0.7451	0.7451
11 NN	0.7647	0.7059	0.8039	0.8431	0.8235	0.8039	0.8039	0.8039	0.8039	0.7843	0.7451	0.7451	0.7451	0.7451
12 NN	0.7843	0.7647	0.7843	0.8431	0.7647	0.7647	0.7647	0.7843	0.7451	0.7059	0.7059	0.7059	0.7059	0.7059
13 NN	0.7843	0.7647	0.7843	0.8431	0.7647	0.7647	0.7647	0.7843	0.7451	0.7059	0.7059	0.7059	0.7059	0.7059
14 NN	0.8039	0.7059	0.7255	0.7843	0.7647	0.7255	0.7647	0.7647	0.7255	0.7255	0.7059	0.7059	0.7059	0.7059
15 NN	0.8039	0.7059	0.7255	0.7843	0.7647	0.7255	0.7647	0.7647	0.7255	0.7255	0.7059	0.7059	0.7059	0.7059

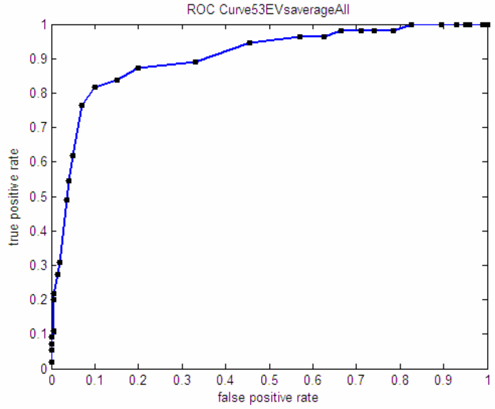
Table 3. Fused recognition rates given number of nearest neighbors and number of eigenvectors.

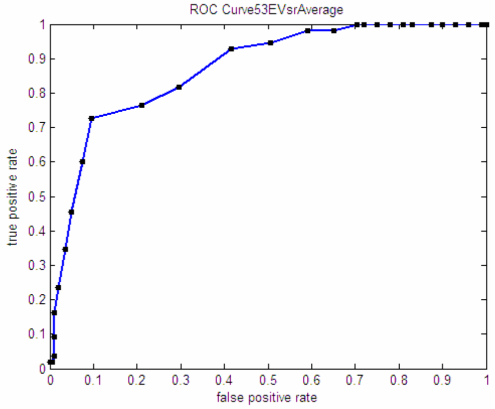
APPENDIX B: ROC CURVES FOR DETECTION

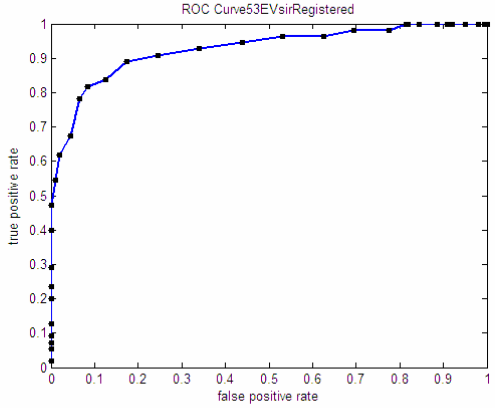
Note: The title of each graph describes first the number of EVs used to create it, 53EVs for instance, and then the type of fusion, gAverage or gReplaced for example. These graphs refer to the detection of vehicles as shown and discussed in Chapter IV of this text.

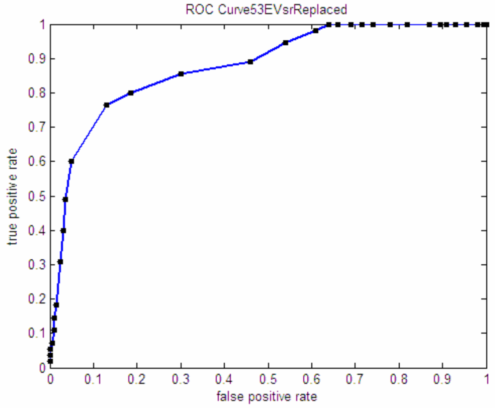


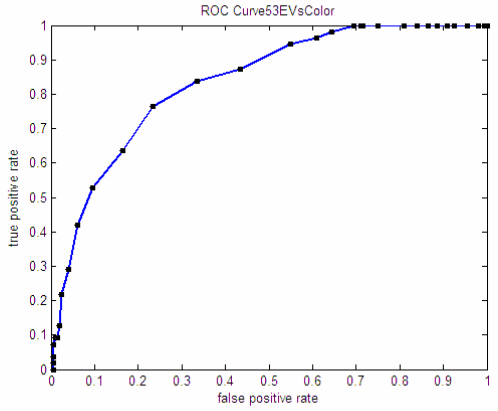


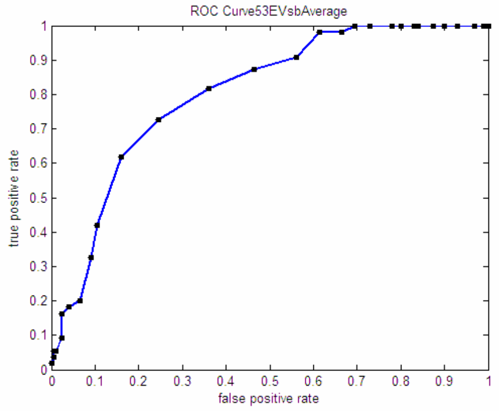


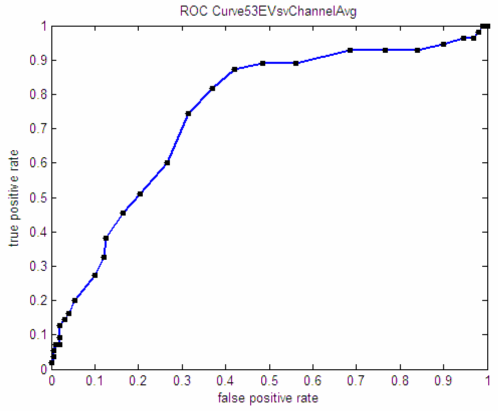


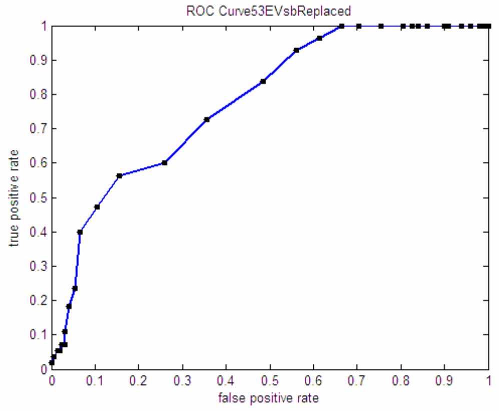


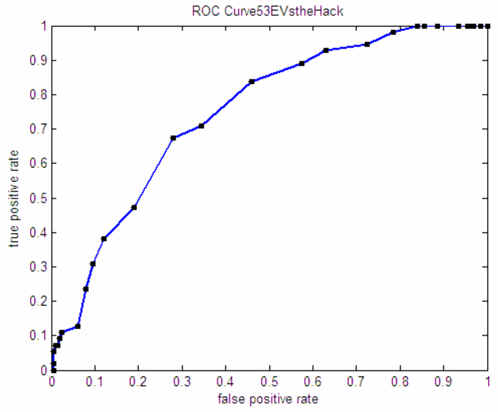


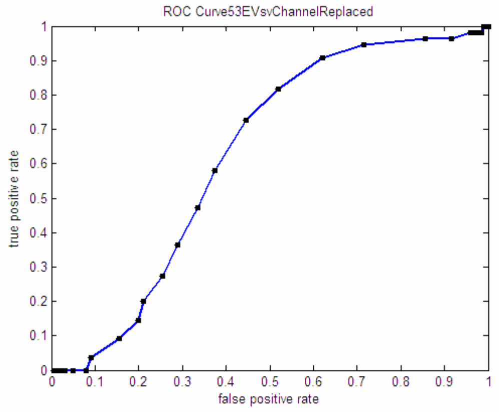

ROC Curve53EVsColor
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC Curve53EVsbAverage
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC Curve53EVsvChannelAvg
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

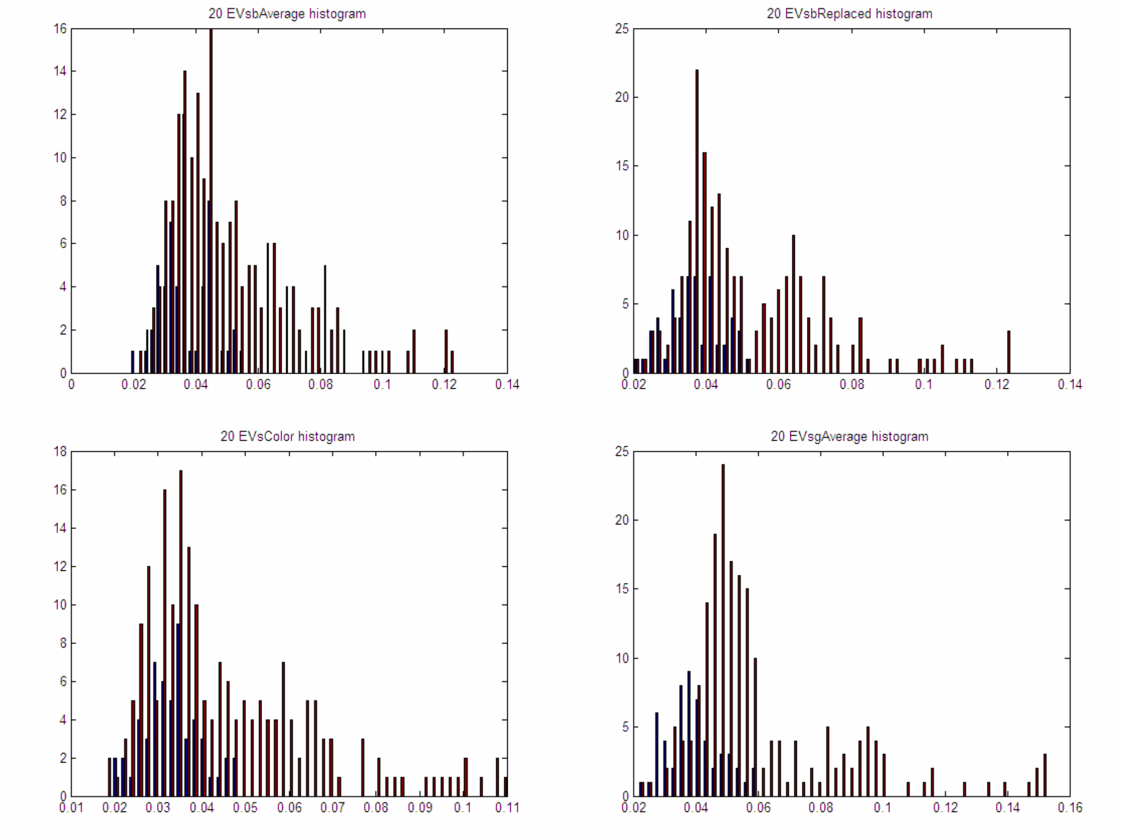

ROC Curve53EVsbReplaced
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

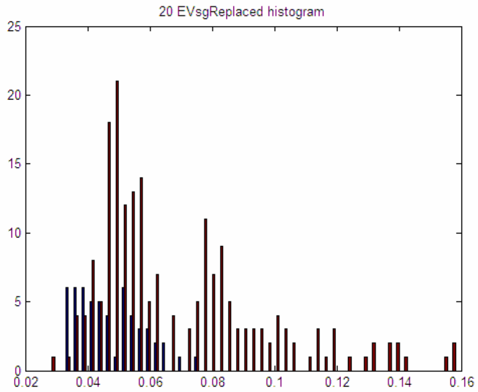

ROC Curve53EVstheHack
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

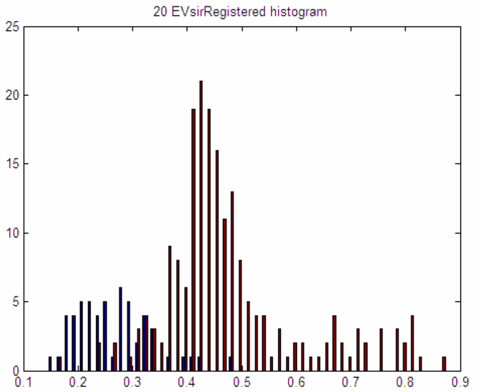

ROC Curve53EVsvChannelReplaced
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

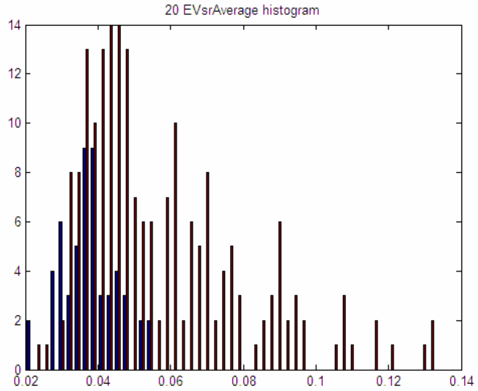
APPENDIX C: HISTOGRAMS OF POSITIVE AND NEGATIVE DISTANCES IN DETECTION

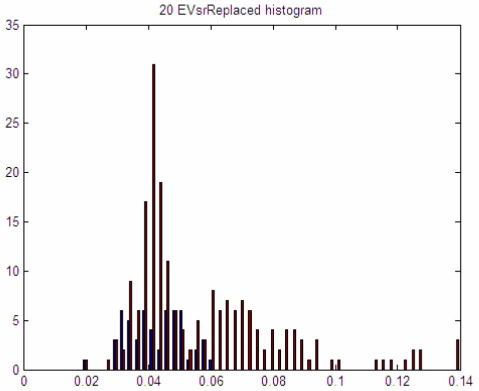
Note that when two columns are present, the first column is for positive distances, the second is for negative distances. Positive distance columns are blue, negative distance columns are red. These histograms refer to the distances from the origin of eigenspace to each positive and negative example as shown and discussed in Chapter IV.

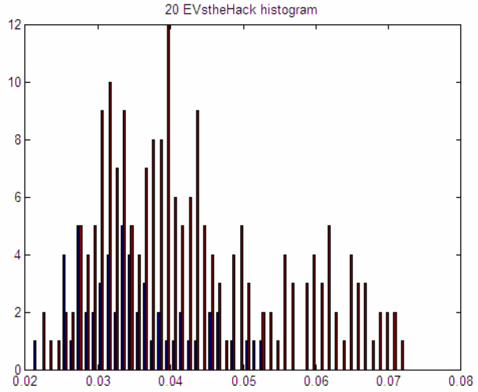


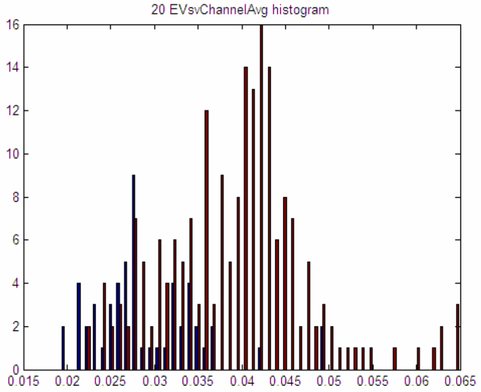

20 EVsgReplaced histogram
0
5
10
15
20
25
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16

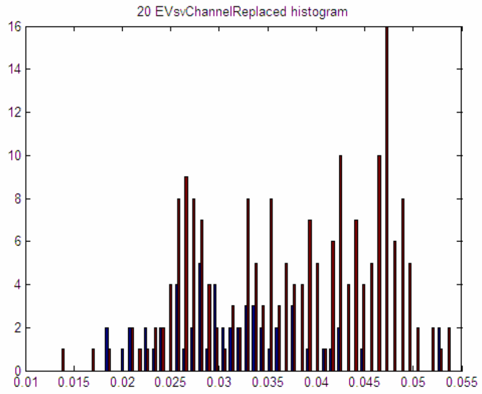

20 EVsirRegistered histogram
0
5
10
15
20
25
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

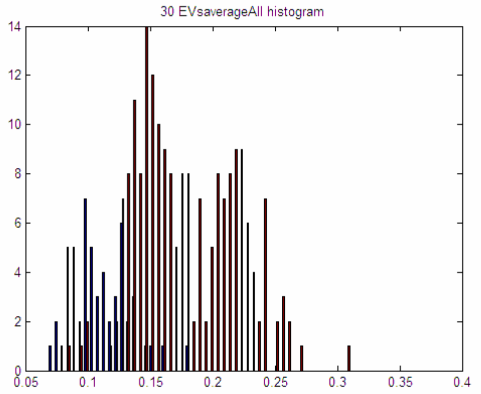

20 EVsrAverage histogram
0
2
4
6
8
10
12
14
0.02
0.04
0.06
0.08
0.1
0.12
0.14

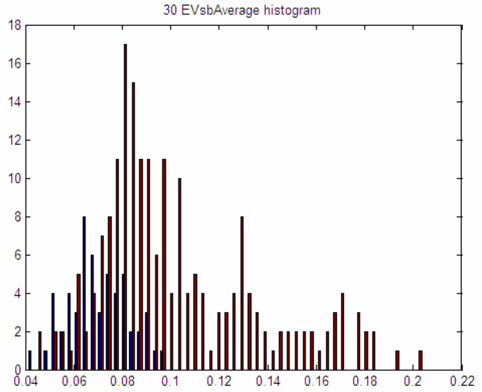

20 EVsrReplaced histogram
0
5
10
15
20
25
30
35
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14

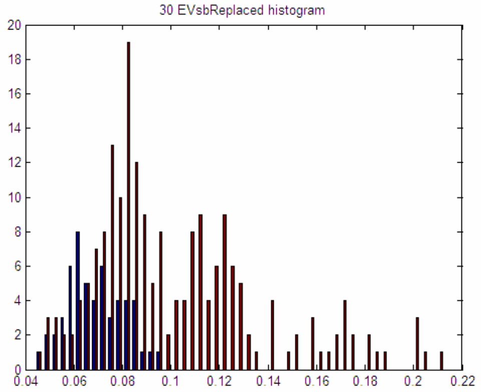

20 EVstheHack histogram
0
2
4
6
8
10
12
0.02
0.03
0.04
0.05
0.06
0.07
0.08

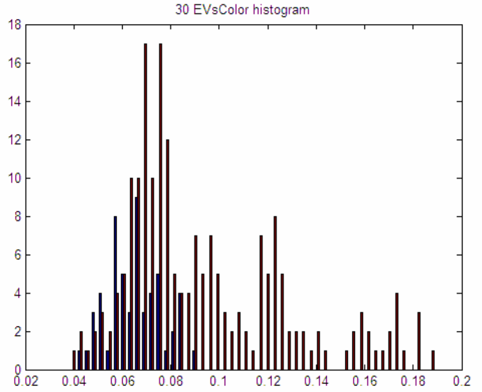

20 EVsvChannelAvg histogram
0
2
4
6
8
10
12
14
16
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
0.055
0.06
0.065

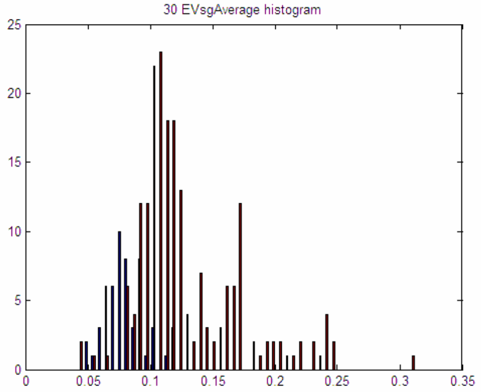

20 EVsvChannelReplaced histogram
16
14
12
10
8
6
4
2
0
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
0.055

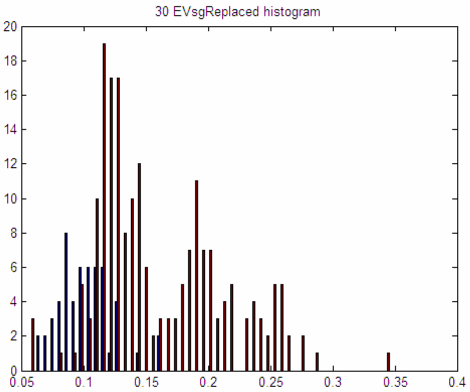

30 EVsaverageAll histogram
14
12
10
8
6
4
2
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4

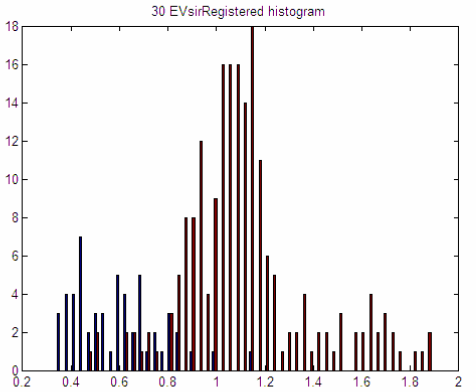

30 EVsbAverage histogram
18
16
14
12
10
8
6
4
2
0
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22

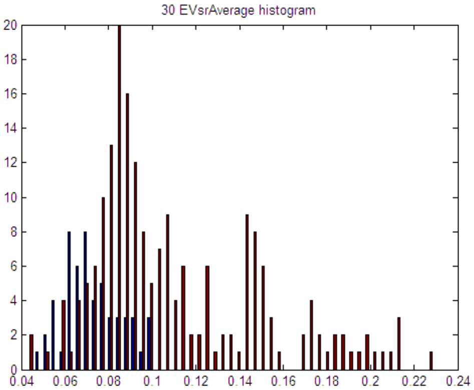

30 EVsbReplaced histogram
20
18
16
14
12
10
8
6
4
2
0
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22

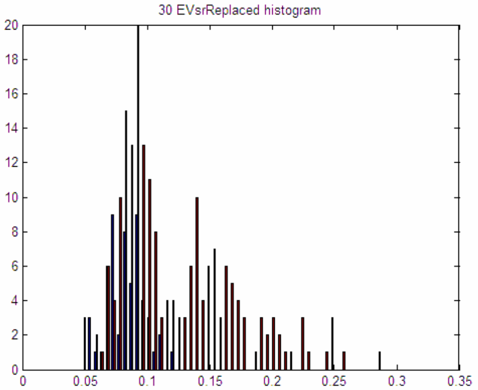

30 EVsColor histogram
18
16
14
12
10
8
6
4
2
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2

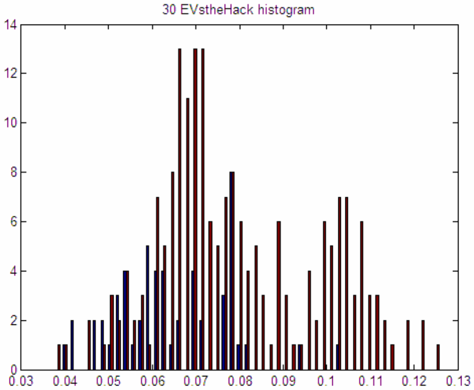

30 EVsgAverage histogram
25
20
15
10
5
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35

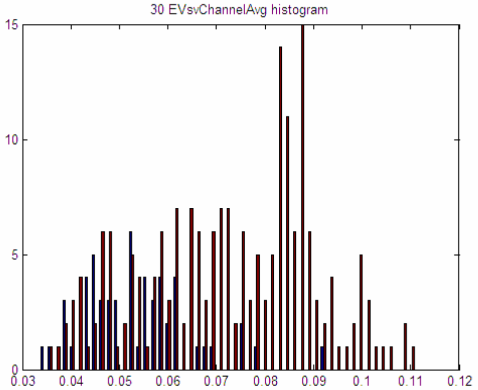

30 EVsgReplaced histogram
0
2
4
6
8
10
12
14
16
18
20
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4

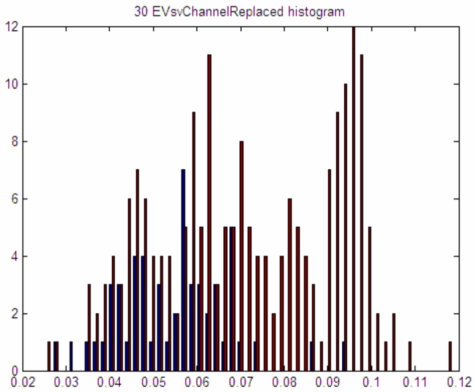

30 EVsirRegistered histogram
0
2
4
6
8
10
12
14
16
18
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


30 EVsrAverage histogram
0
2
4
6
8
10
12
14
16
18
20
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22
0.24

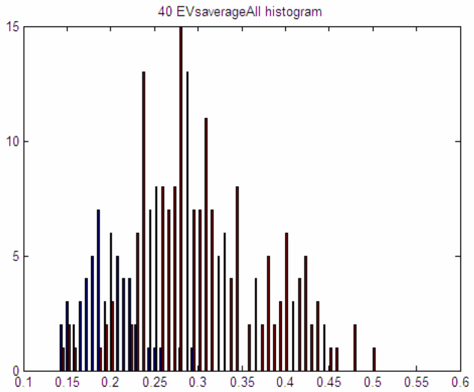

30 EVsrReplaced histogram
0
2
4
6
8
10
12
14
16
18
20
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35


30 EVstheHack histogram
0
2
4
6
8
10
12
14
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
0.11
0.12
0.13

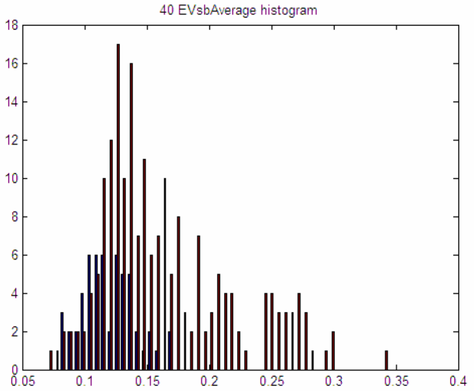

30 EVsvChannelAvg histogram
0
5
10
15
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
0.11
0.12



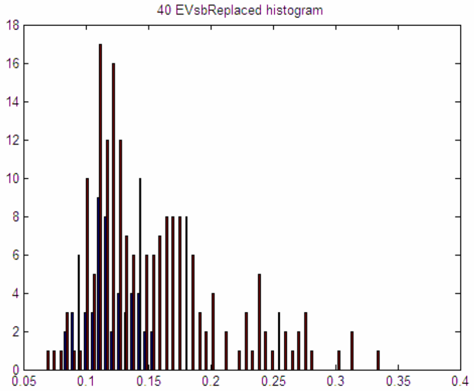
30 EVsvChannelReplaced histogram



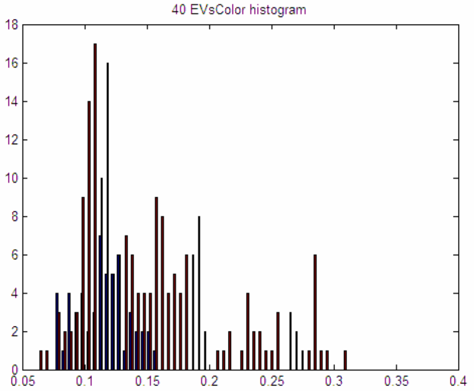
40 EVsaverageAll histogram



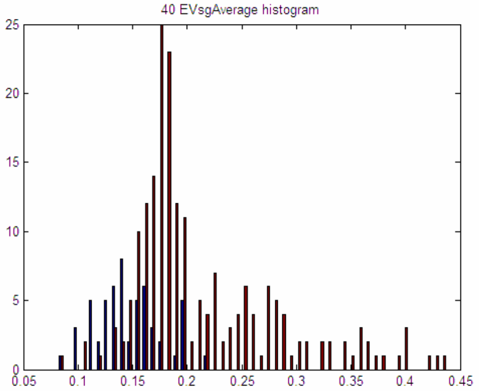
40 EVsbAverage histogram



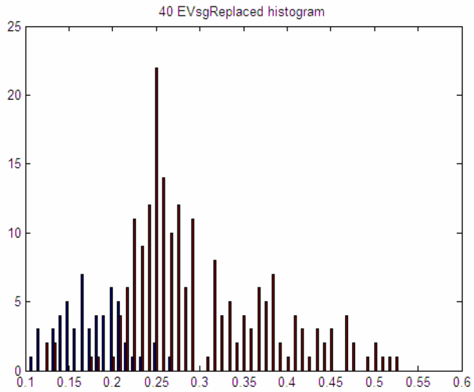
40 EVsbReplaced histogram

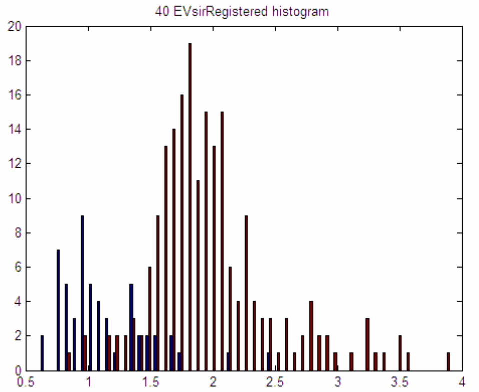


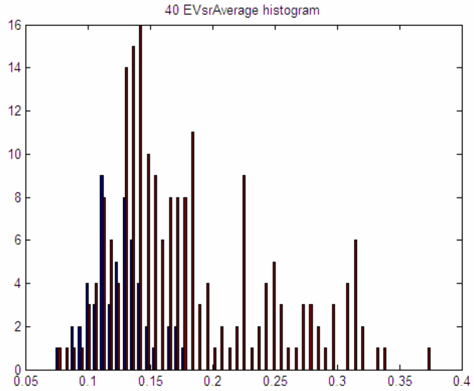
40 EVsColor histogram

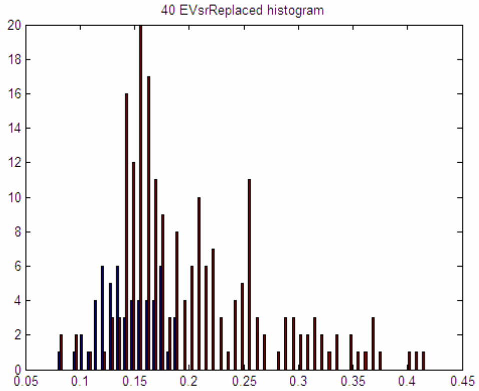


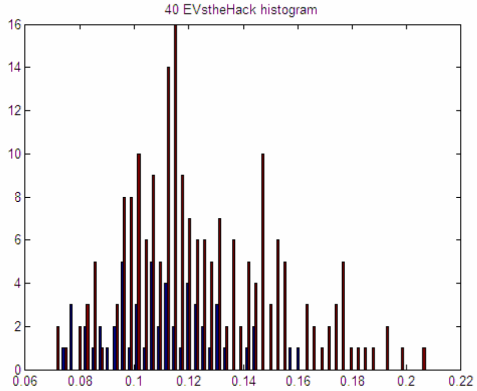
40 EVsgAverage histogram

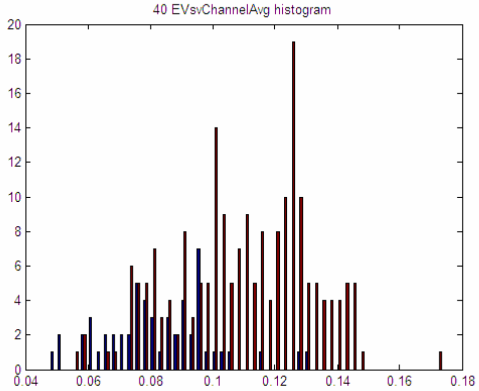

40 EVsgReplaced histogram
25
20
15
10
5
0
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6

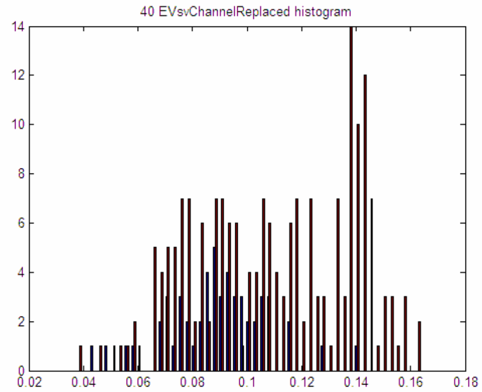

40 EVsirRegistered histogram
20
18
16
14
12
10
8
6
4
2
0
0.5
1
1.5
2
2.5
3
3.5
4

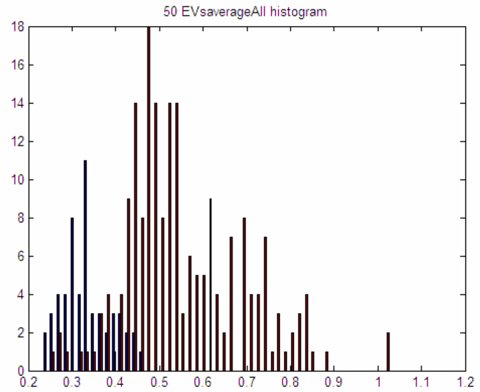

40 EVsrAverage histogram
16
14
12
10
8
6
4
2
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4

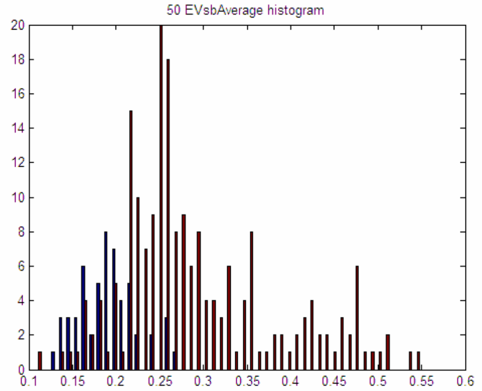

40 EVsrReplaced histogram
20
18
16
14
12
10
8
6
4
2
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45

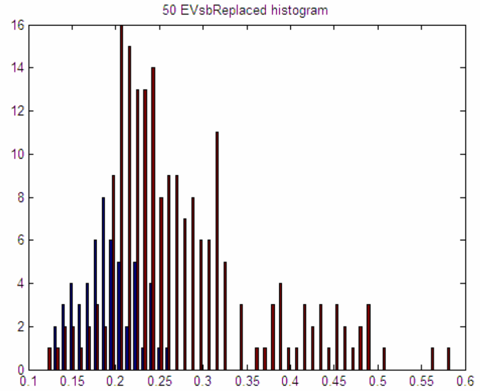

40 EVstheHack histogram
16
14
12
10
8
6
4
2
0
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22

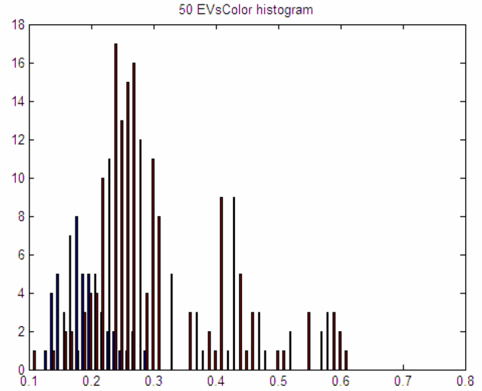

40 EVsvChannelAvg histogram
20
18
16
14
12
10
8
6
4
2
0
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18

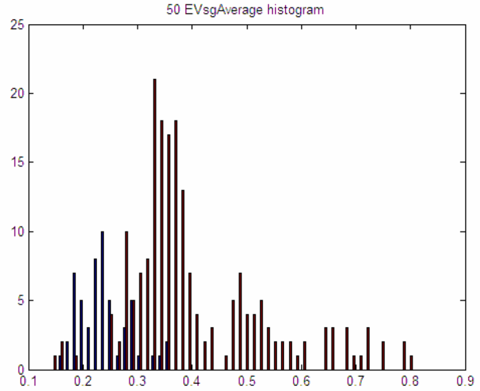

40 EVsvChannelReplaced histogram
0
2
4
6
8
10
12
14
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18

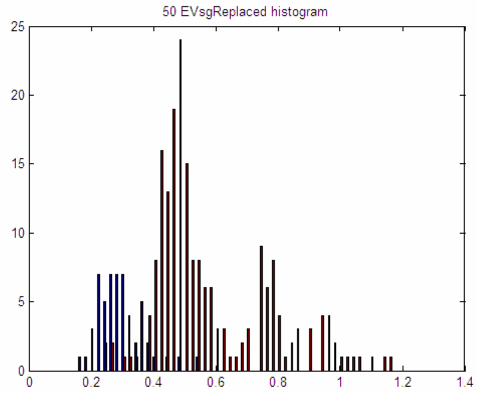

50 EVsaverageAll histogram
0
2
4
6
8
10
12
14
16
18
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2

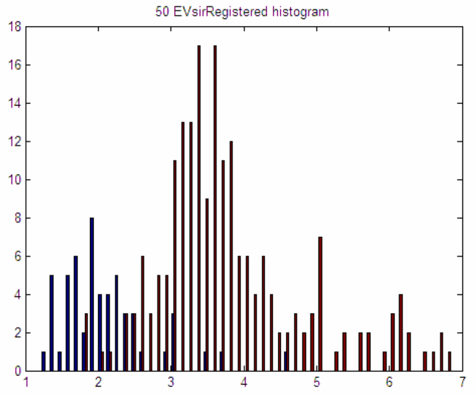

50 EVsbAverage histogram
0
2
4
6
8
10
12
14
16
18
20
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6

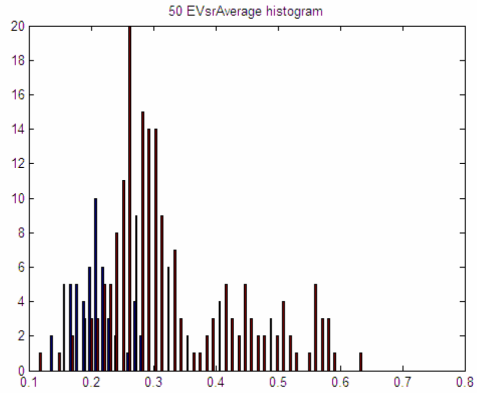

50 EVsbReplaced histogram
0
2
4
6
8
10
12
14
16
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6

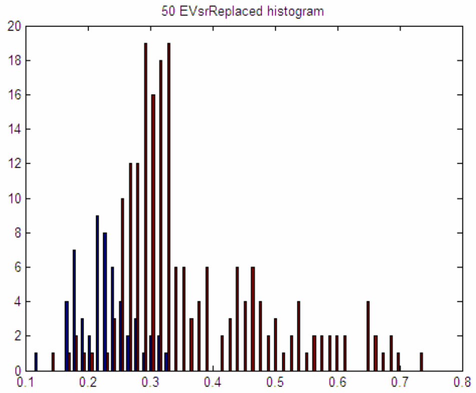

50 EVsColor histogram
0
2
4
6
8
10
12
14
16
18
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

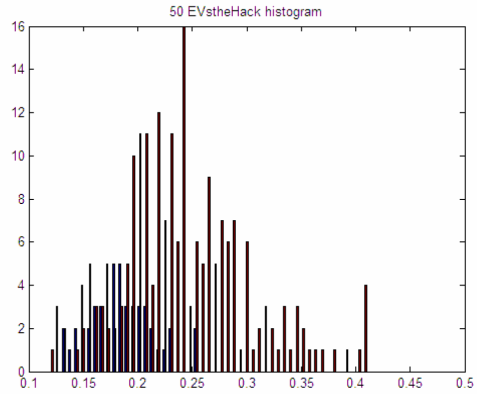

50 EVsgAverage histogram
0
5
10
15
20
25
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

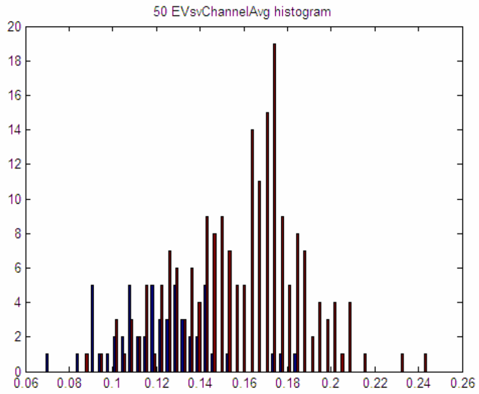

50 EVsgReplaced histogram
0
5
10
15
20
25
0
0.2
0.4
0.6
0.8
1
1.2
1.4

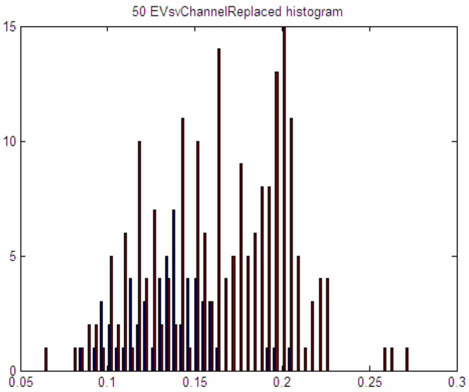

50 EVsirRegistered histogram
0
2
4
6
8
10
12
14
16
18
1
2
3
4
5
6
7

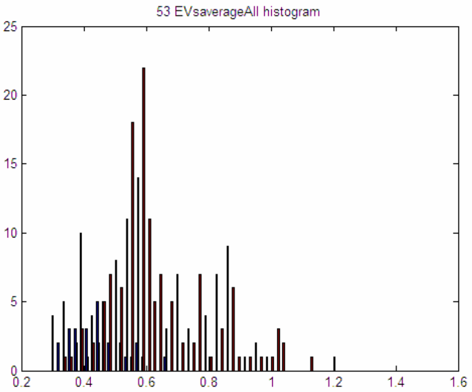

50 EVsrAverage histogram
0
2
4
6
8
10
12
14
16
18
20
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

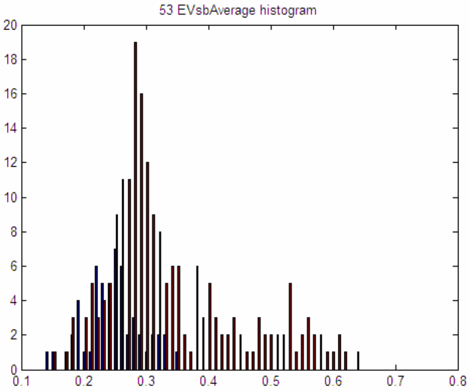

50 EVsrReplaced histogram
0
2
4
6
8
10
12
14
16
18
20
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

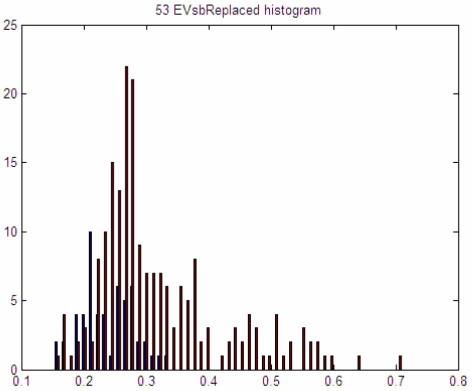

50 EVstheHack histogram
0
2
4
6
8
10
12
14
16
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

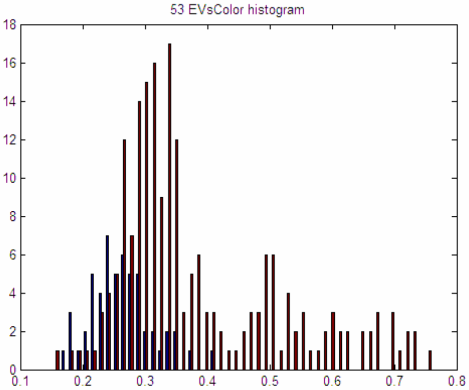

50 EVsvChannelAvg histogram
0
2
4
6
8
10
12
14
16
18
20
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22
0.24
0.26

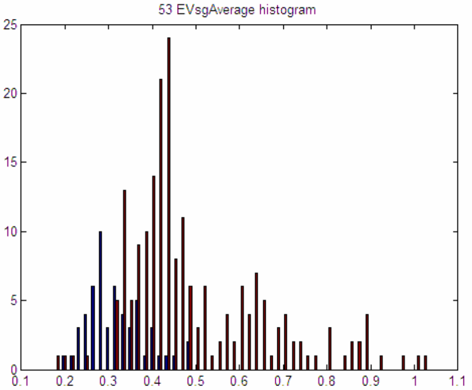

50 EVsvChannelReplaced histogram

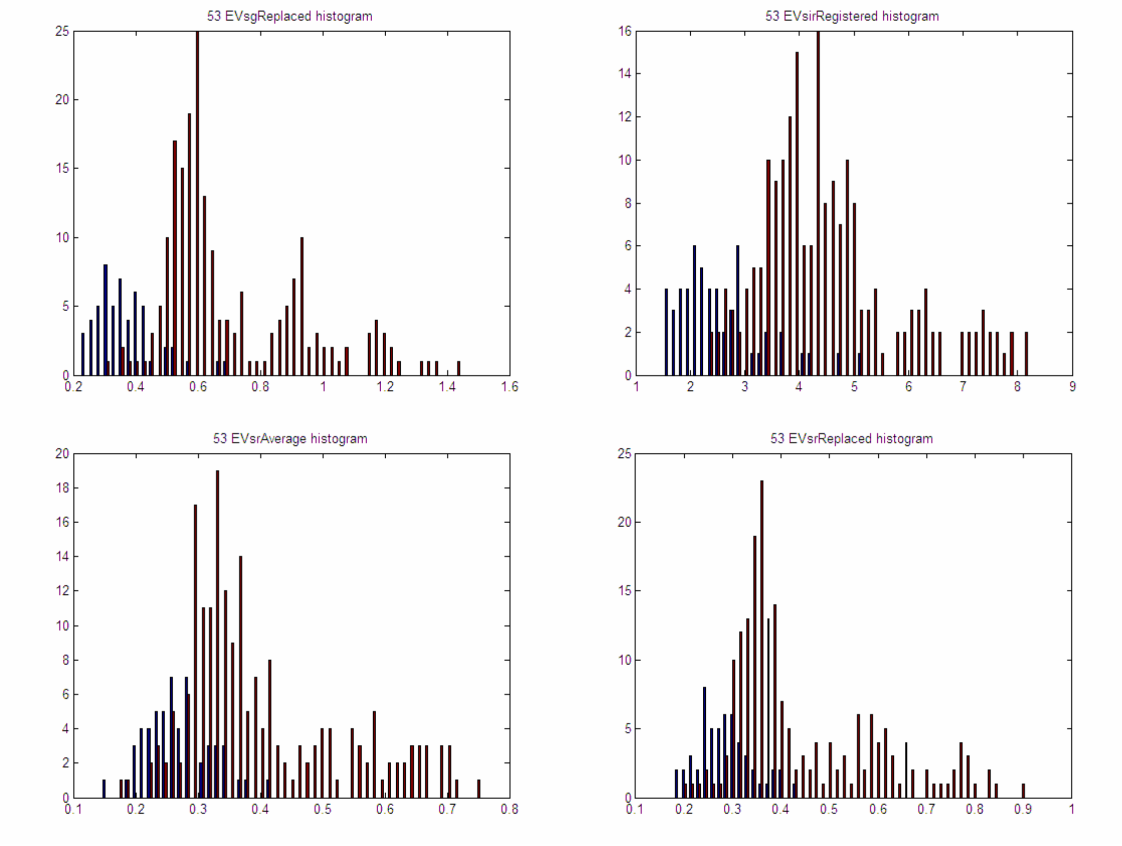

53 EVsaverageAll histogram

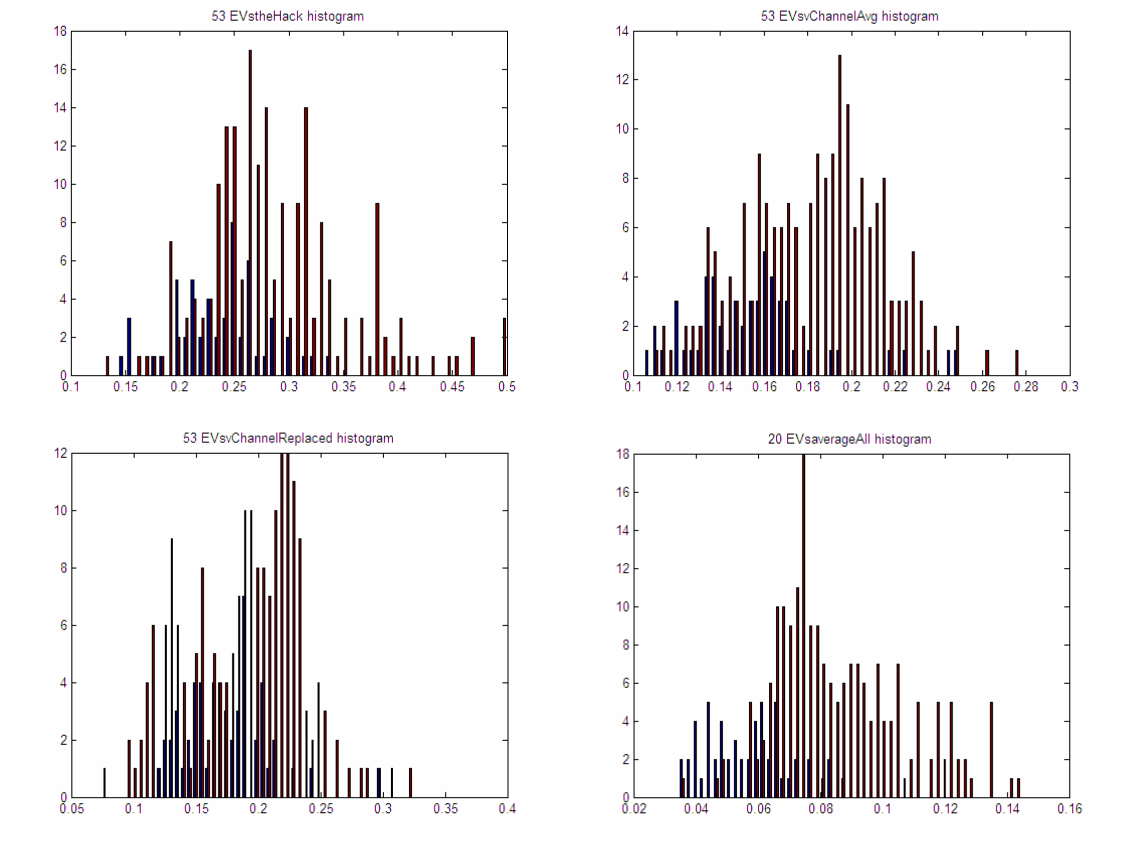

53 EVsbAverage histogram


53 EVsbReplaced histogram


53 EVsColor histogram


53 EVsgAverage histogram


53 EVsgReplaced histogram
0
5
10
15
20
25
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
53 EVsirRegistered histogram
0
2
4
6
8
10
12
14
16
1
2
3
4
5
6
7
8
9
53 EVsrAverage histogram
0
2
4
6
8
10
12
14
16
18
20
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
53 EVsrReplaced histogram
0
5
10
15
20
25
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


53 EVstheHack histogram
0 2 4 6 8 10 12 14 16 18
0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5
53 EVsvChannelAvg histogram
0 2 4 6 8 10 12 14
0.1 0.12 0.14 0.16 0.18 0.2 0.22 0.24 0.26 0.28 0.3
53 EVsvChannelReplaced histogram
0 2 4 6 8 10 12
0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4
20 EVsaverageAll histogram
0 2 4 6 8 10 12 14 16 18
0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Andreone, L. et al. “Vehicle Detection and Localization in Infra-Red Images.” Proceedings of the IEEE Fifth International Conference on Intelligent Transportation Systems, 2002.

Betke, Margrit, Esin Haritaoglu, and Larry S. Davis. “Real-Time Multiple Vehicle Detection and Tracking from a Moving Vehicle.” Machine Vision and Applications 12.2 (2000): 69.

Brown, Lisa Gottesfeld. “A Survey of Image Registration Techniques.” ACM Computing Survey. 24.4 (1992): 325-76.

Brown, Lisa M. “View Independent vehicle/person Classification.” VSSN ‘04: Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks. New York, NY, USA, 2004.

Buluswar, Sashi D., and Bruce A. Draper. “Color Recognition in Outdoor Images.” 1998.

Castleman, Kenneth R. Digital Image Processing. Englewood Cliffs, NJ: Prentice Hall, 1996.

Cramer, H., U. Scheunert and C. Wanielik. “Multi Sensor Fusion for Object Detection using Generalized Feature Models.” Proceedings of the Sixth International Conference of Information Fusion, 2003.

Der, S. Z., and R. Chellappa. “Probe-Based Automatic Target Recognition in Infrared Imagery.” IEEE Transactions on Image Processing, 6.1 (1997): 92-102.

Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern Classification. 2nd ed. New York: Wiley, 2001.

Elachi, Charles, and Jakob Van Zyl. Introduction to the Physics and Techniques of Remote Sensing. 2nd ed. Hoboken, N.J.: Wiley-Interscience, 2006.

“*FM 3-50 Smoke Operations*. 4 December 1990.”

Forsyth, David, and Jean Ponce. Computer Vision: A Modern Approach. Upper Saddle River, NJ: London: Prentice Hall, 2003.

Giachetti, A., M. Campani, and V. Torre. “The use of Optical Flow for Road Navigation.” IEEE Transactions on Robotics and Automation, 14.1 (1998): 34-48.

Hall, David L. and Sonya A.H. McMullen. Mathematical Techniques in Multisensor Data Fusion. Boston: Artech House, 2004.

Henini, Mohamed, and M. Razeghi. Handbook of Infrared Detection Technologies. New York: Elsevier Advanced Technology, 2002.

Hunke, Martin, and Alex Waibel. “Face Locating and Tracking for Human-Computer Interaction.” Conference record of the 28th Asilomar Conference on Signals, Systems & Computers, Pacific Grove, Calif. A (1994.)

Holst, Gerald C. Common Sense Approach to Thermal Imaging. Winter Park: JCD Publishing, 2000.

Jha, A.R. Infrared Technology: Applications to Electrooptics, Photonic Devices and Sensors. New York: Wiley, 2000.

Jolliffe, I.T. Principal Component Analysis. New York: Spring-Verlag, 2002.

Lowe, David G. “Object Recognition from Local Scale-Invariant Features.” ICCV ‘99: Proceedings of the International Conference on Computer Vision-Volume 2.

Kagesawa, M., et al. “Recognizing Vehicles in Infrared Images using IMAP Parallel Vision Board.” IEEE Transactions on Intelligent Transportation Systems, 2.1 (2001): 10-7. .

Kaplan, Herbert. Practical Applications of Infrared Thermal Sensing and Imaging Equipment. Bellingham: SPIE, 2007.

Ke, Yan, and R. Sukthankar. “PCA-SIFT: A More Distinctive Representation for Local Image Descriptors.” Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.

Kokar, Mieczyslaw M., and Jerzy A. Tomasik. Data Vs. Decision Fusion in the Category Theory., 2001.

Kruse, Paul W. Uncooled Thermal Imaging: Arrays, Systems, and Applications. Bellingham: SPIE, 2001.

Kuehnle, Andreas. “Symmetry-Based Recognition of Vehicle Rears.” Pattern Recognition Letters. 12.4 (1991): 249-58.

Martin, Christian, et al. “Sensor Fusion using a Probabilistic Aggregation Scheme for People Detection and Tracking.” Proceedings of the Second European Conference on Mobile Robots (ECMR 2005) (2005): 176.

Ming-Hsuan Yang, D. J. Kriegman, and N. Ahuja. “Detecting Faces in Images: A Survey.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 24.1 (2002): 34-58.

Mitianoudis, N., and Stathaki, T. “Adaptive Image Fusion using Ica Bases.”

Nandhakumar, N. “A Phenomenological Approach to Multisource Data Integration: Analyzing Infrared and Visible Data.” Proceeding of the IAPR TC7 Workshop on Multisource Data Integration in Remote Sensing, College Park, MD, June 14-15, 1990.

Nelson, B. N. “Automatic Vehicle Detection in Infrared Imagery using a Fuzzy Inference-Based Classification System.” IEEE Transactions on Fuzzy Systems, 9.1 (2001): 53-61.

Nett, E., and S. Schemmer. “Realizing Virtual Sensors by Distributed Multi-Level Sensor Fusion.” Proceedings of Second International Work shop on Multi-Robot Systems (2003).

Olsson, L., C. L. Nehaniv and D. Polani. “Sensor Adaptation and Development in Robots by Entropy Maximization of Sensory Data.” Proceedings of the 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2005. CIRA 2005.

Piella, G. “A Region-Based Multiresolution Image Fusion Algorithm.”

Sun, Z., G. Bebis and R. Miller. “On-Road Vehicle Detection using Optical Sensors: A Review.” IEEE International Conference on Intelligent Transportation Systems, 2004.

Tso, B. and Mather, P.M. Classification Methods for Remotely Sensed Data. London: Taylor and Francis, 2001.

Turk, M. “Eigenfaces and beyond,” in W. Zhao and R. Chellappa (eds.), “Face Processing: Advanced Modeling and Methods.” Academic Press, 2005.

Turk, M. A., and A. P. Pentland. “Face Recognition using Eigenfaces.” Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. CVPR ‘91.

Webb, A. R. Statistical Pattern Recognition. 2nd ed. West Sussex, England: New Jersey: Wiley, 2002.

Yilmaz, Alper. “Sensor Fusion in Computer Vision.”

Zhao, W. et al. “Face Recognition: A Literature Survey.” ACM Computing.Survey. 35.4 (2003): 399-458.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California